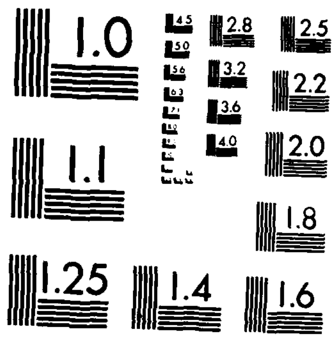
AD-A141 386 LCAP2 (LINEAR CONTROLS ANALYSIS PROGRAM) VOLUME 3 1/4
SOURCE CODE DESCRIPTION..(U) AEROSPACE CORP EL SEGUNDO
CA GUIDANCE AND CONTROL DIV E A LEE 15 NOV 83
UNCLASSIFIED TR-0084(9975)-1-VOL-3 SD-TR-84-06-VOL-3 F/G 9/2 NL


1.0
1.1
1.25
1.4
1.6
1.8
2.0
2.2
2.5
2.8
3.2
3.6
4.0

REPORT SD-TR-84-08, Vol. III

12

# LCAP2 - Linear Control Analysis Program

## Volume III: Source Code Description

E. A. LEE
Guidance and Control Division
Engineering Group
The Aerospace Corporation
El Segundo, Calif. 90245



15 November 1983

Final Report



APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED



Prepared for

SPACE DIVISION
AIR FORCE SYSTEMS COMMAND
Los Angeles Air Force Station
P.O. Box 92960, Worldway Postal Center
Los Angeles, California 90009

This report was submitted by The Aerospace Corporation, El Segundo, CA 90245, under Contract No. F04701-83-C-0084 with the Space Division, P.O. Box 92960, Worldway Postal Center, Los Angeles, CA 90009. It was reviewed and approved for The Aerospace Corporation by Ronald G. Nishinaga, Advanced Programs, Defense Development Division. Major William J. Cooper, SD/YDS, was the Air Force project officer.

This report has been reviewed by the Public Affairs Office (PAS) and is releasable to the National Technical Information Service (NTIS). At NTIS, it will be available to the general public, including foreign nationals.

This technical report has been reviewed and is approved for publication. Publication of this report does not constitute Air Force approval of the report's findings or conclusions. It is published only for the exchange and stimulation of ideas.

William J. Cooper
Project Officer

Joseph Hess
Joseph Hess, GM-15, Director
West Coast Office, Air Force
Space Technology Center

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

# REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

| | |
|---|---|
| 1. REPORT NUMBER: SD-TR-84-06, Vol. III | 2. GOVT ACCESSION NO.: AI41386 |
| 3. RECIPIENT'S CATALOG NUMBER | |
| 4. TITLE (and Subtitle): LCAP2 - LINEAR CONTROL ANALYSIS PROGRAM Volume III: Source Code Description | 5. TYPE OF REPORT & PERIOD COVERED: Final Report |
| | 6. PERFORMING ORG. REPORT NUMBER: TR-0084(9975)-1, Vol. III |
| 7. AUTHOR(s): Eugene A. Lee | 8. CONTRACT OR GRANT NUMBER(s): F04701-83-C-0084 |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS: The Aerospace Corporation El Segundo, Calif. 90245 | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS |
| 11. CONTROLLING OFFICE NAME AND ADDRESS: Space Division Los Angeles Air Force Station Los Angeles, Calif. 90009 | 12. REPORT DATE: 15 November 1983 |
| | 13. NUMBER OF PAGES: 307 |
| 14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) | 15. SECURITY CLASS. (of this report): Unclassified |
| | 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE |

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Linear systems
Digital control systems
Frequency response
Root locus
Laplace transform
Sampled data
z-transform
w-transform
Multirate sampling
Frequency decomposition

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The computer program LCAP2 (Linear Controls Analysis Program) provides the analyst with the capability to numerically perform classical linear control analysis techniques such as transfer function manipulation, transfer function evaluation, frequency response, root locus, time response and sampled-data transforms. It is able to deal with continuous and sampled-data systems, including multiloop multirate digital systems, using s, z and w transforms.

DD FORM 1473 (FACSIMILE)

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

19. KEY WORDS (Continued)

Control systems analysis program
Inverse transforms
Cramer's method
Transfer function evaluation

20. ABSTRACT (Continued)

Primary considerations in the development of this program were ease of use and computational accuracy. Transfer function and polynomial arrays are defined to be referenced with indices so that they may be easily addressed by the operators. The combination of this set of LCAP2 operators and the form of the data structure provides a very flexible and easy to use program.

Since each LCAP2 operator is coded as a FORTRAN subroutine, the batch version of LCAP2 allows the user to easily develop code to automate, for example, a complete stability analysis task beginning with the input of raw data to the generation of the stability plots. An interactive version of LCAP2 is also available.

The LCAP2 report is organized in three volumes: batch user's guide (I), interactive user's guide (II), and source code description (III).

## PREFACE

This is the first issue of the LCAP2 source code description report. Description of this program still needs to cover several aspects of the program before this report can be considered as complete. These include description of (1) blank and labeled COMMON block, (2) flow diagrams for typical Batch and Interactive LCAP2 operators, and (3) methods for selective suppression of print-outs. Rather than wait until time is available to complete this task, this report is published in its present state so that it can be referenced by LCAP2 users.

CONTENTS


1.0 INTRODUCTION . . . . . 1
2.0 PROGRAMMING CONSIDERATIONS . . . . . 1
3.0 BATCH LCAP2 . . . . . 2
3.1 JOB STRUCTURE . . . . . 2
3.2 JOB SUBMITTAL . . . . . 3
4.0 INTERACTIVE LCAP2 . . . . . 6
4.1 JOB STRUCTURE FOR CREATION OF INTERACTIVE LCAP2 BINARY CODE . . . . 6
4.2 JOB SUBMITTAL FOR CREATION OF INTERACTIVE LCAP2 BINARY CODE . . . . 7
4.3 PROC FOR LOADING AND EXECUTING INTERACTIVE LCAP2 . . . . . 9
5.0 LCAP2 DATA FORMAT . . . . . 10
5.1 POLYNOMIAL FORMAT . . . . . 10
5.2 TRANSFER FUNCTION FORMAT . . . . . 12
5.3 MATRIX FORMAT . . . . . 13
5.4 B VECTOR FORMAT . . . . . 14
6.0 SUBROUTINES . . . . . 15
ADDP . . . . . 15
AUXM1 . . . . . 16
AUXM2 . . . . . 18
AUXP . . . . . 20
AUXP1 . . . . . 21
AUXRT1 . . . . . 22
AXXMRN . . . . . 23
BILNWZ . . . . . 24
BILNZW . . . . . 25
BPRINT1 . . . . . 26
BPRINT2 . . . . . 27
CNGCO . . . . . 28
CNGCOP . . . . . 29
CNGRT . . . . . 30
CNGRTP . . . . . 31
COEFF . . . . . 32
COEFP . . . . . 33
CPPRN . . . . . 34
CPYPS . . . . . 35
CPYPW . . . . . 36
CPYPZ . . . . . 36
CPYSP . . . . . 37
CPYWP . . . . . 38
CPYZP . . . . . 38
CRELIM . . . . . 39
CXMTX1 . . . . . 40
DAYPRN . . . . . 41
DETRM . . . . . 42
DTERM . . . . . 44
DOTLINE . . . . . 46
EJKN . . . . . 47

EJKNI . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 48
ELPLOT1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 49
ENDLINE . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 51
EVLRRT . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 52
EVLRT . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 53
EVMRRT . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 54
FAUXW . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 55
FAUXWM . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 56
FAUXZ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 57
FAUXZM . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 58
FCNPLN . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 59
FCNW1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 60
FCNW2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 61
FETPY . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 62
FETSTF . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 63
FETTFX . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 64
FETWTF . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 65
FETZTF . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 67
FPLOT . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 69
FPLOT1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 70
FREQS . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 72
FREQSI1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 73
FREQS2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 74
FREQS3 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 75
FREQW . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 76
FREQWM1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 77
FREQWM2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 79
FREQW1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 81
FREQW2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 83
FREQZ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 85
FREQZM1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 86
FREQZM2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 88
FREQZ1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 90
FREQZ2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 92
GRAF1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 94
HEADINi,(i=1,5) . . . . . . . . . . . . . . . . . . . . . . . . . . 95
HELP . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 96
HOLLI . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 97
IARG0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 98
IARG1A . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 99
IARG1B . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 100
IARG2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 101
IDETRM . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 102
IDTERM . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 103
ILCAP2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 104
INIT0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 105
INTXFM . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 106
IROOT . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 107
IROOTP . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 108
IRSTOR . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 109

ISTORE . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 110
ITITLE . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 111
LEXIT . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 112
MATROT1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 113
MINIT0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 114
MINIT2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 115
MPRINT1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 116
MPRINT2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 117
MROOT1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 118
MROOT2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 120
MRXFM . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 122
MULE . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 123
NORM . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 125
OPMESG . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 126
OPPRN . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 127
OSCALE . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 129
PACK . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 130
PADD . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 132
PCHEK . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 133
PEQU . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 134
PEQUAL . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 135
PLDC . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 136
PLDR . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 137
PLQ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 138
PMULT . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 139
PMPY . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 140
PPADD . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 141
PPEQU . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 142
PPMPY . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 143
PPRINT . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 144
PPRN . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 145
PPRN1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 146
PPSUB . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 147
PPZERO . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 148
PROOT . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 149
PRTS . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 150
PSUB . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 151
PSYNTH . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 152
PYPRN1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 153
PYPRN4 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 154
PZERO . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 155
RCLAS . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 156
REMARKi,(i=1,5) . . . . . . . . . . . . . . . . . . . . . . . . . 158
RESDU . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 159
RESTORE . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 160
RLOCIN1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 161
RLOCUS1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 162
RPRINT . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 164
RRTADD . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 165
RRTEQU . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 166

RRTSUB . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 167
RRTPRN . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 168
RRZERO . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 169
RTCMNT . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 170
RTEQU . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 171
RTEQ1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 172
RTMPY . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 173
RTPRN . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 174
RTPRN0 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 175
RTPRN2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 176
RZERO . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 177
SELCR . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 178
SFAUX . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 179
SFAUX1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 180
SFREQ . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 182
SFREQY . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 186
SLOCI . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 187
SNORM . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 189
SPADD . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 190
SPDIV . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 191
SPEQU . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 192
SPLDC . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 193
SPLDR . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 194
SPMPY . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 195
SPPRN . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 196
SPRTS . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 197
SPSUB . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 198
STIME . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 199
STIME1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 201
STIME2 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 202
STIME3 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 204
STORE . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 205
STRPY . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 206
STRSTF . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 207
STRTFX . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 208
STRWTF . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 209
STRZTF . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 209
SUBP . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 210
SWMRX . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 211
SWXFM . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 213
SZMRX . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 215
SZXFM . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 217
TFAUX . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 218
TFPRN1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 219
TFPRN4 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 220
TSPLOT . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 221
USRNOTE . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 222
WELCR . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 223
WFAUX . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 224
WFAUX1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 225

WFREQ . . . 227
WFREQY . . . 228
WLOCI . . . 229
WMFAUX . . . 230
WMRFQ . . . 231
WMRFRQY . . . 232
WMRTRAN . . . 233
WMRXFM . . . 234
WNORM . . . 236
WPADD . . . 237
WPDIV . . . 238
WPEQU . . . 239
WPLDC . . . 240
WPLDR . . . 241
WPMPY . . . 242
WPPRN . . . 243
WPRTS . . . 244
WPSUB . . . 245
WSTRAN1 . . . 246
WSXFM . . . 247
WTRANS . . . 248
WZXFM . . . 249
XTRACT . . . 250
ZELCR . . . 251
ZEQ1 . . . 252
ZETAZXM . . . 253
ZFAUX . . . 254
ZFAUX1 . . . 255
ZFREQ . . . 257
ZFREQY . . . 258
ZHOLLI . . . 259
ZLOCI . . . 260
ZMFAUX . . . 261
ZMRFQ . . . 262
ZMRFRQY . . . 263
ZMRXFM . . . 264
ZNORM . . . 265
ZPADD . . . 266
ZPDIV . . . 267
ZPEQU . . . 268
ZPLDC . . . 269
ZPLDR . . . 270
ZPMPY . . . 271
ZPPRN . . . 272
ZPRTS . . . 273
ZPSUB . . . 274
ZSXFM . . . 275
ZTIME . . . 276
ZTIME1 . . . 278
ZTIME2 . . . 279

ZVCHNG1 . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 280
ZVCNG . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 281
ZWTRANS . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 282
ZWXFM . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 283
ZZETAXM . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 284

REFERENCES . . . . . . . . . . . . . . . . . . . . . . . . . . . . 285

Appendix A. COMDECK LCAP2 FOR PROGRAM UPDATE . . . . . . . . . . . . 287

Appendix B. SEGLOAD DIRECTIVES FOR LOADING INTERACTIVE LCAP2 ROUTINES 39

Appendix C. PROCEDURE INTLCAP2 FOR INTERACTIVE LCAP2 . . . . . . . . . . 293

Appendix D. PROGRAM GENPROC FOR INTERACTIVE LCAP2 . . . . . . . . . . . 295

## 1.0 INTRODUCTION

LCAP2 (Linear Control Analysis program) is a FORTRAN program which provides the control analyst with the capability to numerically perform classical linear control analysis techniques such as transfer function manipulation, transfer function evaluation, frequency response, root locus, inverse time response and sampled-data transforms, including multiloop multirate digital systems, using s, z and w transforms.

This program is the successor to LCAP, Ref. 1, which is a batch program utilizing card inputs. This original version did not have the flexibility to allow the user to easily develop code to automate, for example, a complete stability analysis task beginning with the input of raw data to the generation of the stability plots. This is a very desirable feature in an industrial environment. The batch version of LCAP2 provides this flexibility, since the user writes his own FORTRAN program and each LCAP2 operator is coded as a FORTRAN subroutine. The interactive version of LCAP2 is also a very easy program to use since numerous prompts are employed. However, it is not as flexible as the batch version since the user can only input numeric and simple alphanumeric data. The interactive version is primarily intended to be used for quick response design and analysis of systems which can be easily modeled.

## 2.0 PROGRAMMING CONSIDERATIONS

The original version of LCAP, which was written in the late 1960's, utilized an overlay structure since the existing core memory of the CDC 6400/6600 at that time was not sufficient to load the entire LCAP program. The use of overlays in this original version of the program was of no concern to the user since use of LCAP consisted of loading and executing a binary program.

To provide the user with more flexibility in using this program, a major revision was started in the late 1970's. The improved flexibility was achieved by allowing the user to write his own source code in FORTRAN. Overlays were not used for this improved version of the program for the following reasons: (1) user does not have to contend with understanding an overlay structure, (2) CDC 176 core memory is larger than the previous CDC 6400/6600, and (3) an LCAP2 library was created. Creation of the LCAP2 library allows a user job to load only those routines necessary to execute the job, thereby minimizing the core memory required.

A major effort in the revision of LCAP was the implemention of each LCAP operator as a FORTRAN subroutine so that the user could write a simple FORTRAN CALL statement to specify an LCAP operation. Coding of the fundamental control analysis functions previously implemented in LCAP were not changed. In fact, some of this code, which was developed in the 1960's, could be reworked to improve its efficiency. However, due to limited resources, this was not done.

In 1980 development of the interactive version of LCAP2 was initiated. This version of the program was to provide quick response support for the Aerospace Corporation's design and analysis capabilities. Design of this program was

dictated by the hardware and software provided by the CDC 835 computer using the interactive INTERCOM Version 5.0 system and the NOS Version 1 operating system. In the interactive mode only 120K octal words are available to the user. This is hardly adequate for loading LCAP2. The interactive version of LCAP2 had to use either an overlay structure or employ a segment loader. The latter was chosen since the effort required to define a tree structure for the segment loader was easier than defining an overlay structure. This difference in effort is even more apparent when modifications or additions are to be made to LCAP2 since the segment loader directives, which define the tree structure, are all on one record while overlay statements are intersperced throughout the FORTRAN source code.

In the batch version of LCAP2 where the user writes his own FORTRAN source code, flexibility of this program is provided by utilization of the construction of the FORTRAN language and by the FORTRAN callable implementation of the LCAP2 operators. A similar flexibility for the interactive version of LCAP2 would be desirable. However, since FORTRAN is a compiler language, rather than an interpretive language, an interactive command language is needed to interpret the user's FORTRAN or FORTRAN like statements. Since such a command language was not available and no resources were available to develop one, a less flexible interactive program was developed. By limiting the user inputs to numeric and simple alphanumeric data and by extensive use of user prompts, a versatile and easy to use program was implemented.

## 3.0 BATCH LCAP2

Use of the batch version of LCAP2 consists of the user writing his own FORTRAN program. This allows the user to make full use of the FORTRAN language in developing an analysis program. The principle involved in providing the user with this capability is simple. However, the actual operations involved are a bit more complex. Description of this process is given in two parts: (1) job structure and (2) job submittal.

## 3.1 JOB STRUCTURE

The basic operations of a batch LCAP2 job are:

(1) Creation of FORTRAN program - main program and subroutines (optional)

(2) Compilation of the source code from (1)

(3) Loading of routines from the LCAP2 and system libraries

(4) Execution of the LCAP2 program from (2)

(5) (Optional) - Cataloging of data file if one is created by LCAP2

(6) (Optional) - Loading and execution of the HARDCPY program to produce hardcopy plo . created by LCAP2

To facilitate the development of the FORTRAN program by the user, the CDC UPDATE[1] program is utilized. An LCAP2 library has been defined so that the first part of the main program (see Appendix A), which contains many lines of COMMON block and EQUIVALENCE statements, need not be written by the user. This block of code is copied from the program library and added to the the user's FORTRAN code to create the source code. The input (card images) for the UPDATE program will be of the form:

```
*IDENT idname
*INSERT START.1
*DECK MAIN
*CALL LCAP2
      CALL INITO        (initialization of LCAP2 parameters)
      CALL MINITO       (initialization of matrix parameters)
           .
           .
      (user's FORTRAN code)
           .
           .
      CALL LEXIT        (required if hardcopy plots are generated)
      END
```

The * in column 1 defines an UPDATE directive. The first directive, *IDENT idname, specifies an identification name, idname, which can be 1 through 9 characters long. The second directive, *INSERT START.1, defines the location where the input data to follow is to be inserted. The directive, *CALL LCAP2, will write the code in COMDECK LCAP2 to the file COMPILE. This COMDECK LCAP2 contains the main program statment and all of the COMMON block and EQUIVALENCE statements required by the main program. The remaining input data are the user's FORTRAN code which will be copied to the file COMPILE to complete the creation of the main program.

The job control cards for setting up the above operations are given in the next section.

## 3.2 JOB SUBMITTAL

Two forms for job submittal are given below. The first will be an explicit one which includes a complete list of control cards required. The second is a shortened form which attaches and uses a procedure to generate the control cards.

[1] The UPDATE program maintains and updates source decks for libraries under the SCOPE 2.1, NOS 1, and NOS/BE 1 operating systems.

The *first form* is:

```
------------------------------------------------------------------------------
                 .
(    control cards for accounting   )
                 .
FILE,TAPE30,BT=I.                           (optional, use only if old data
ATTACH(TAPE30,lfn,ID=.....,ST=PF6)           is to be restored)
ATTACH(OLDPL,8LCAP2PLX,ID=9487)             (attach LCAP2 program library )
UPDATE.
FTN(I=COMPILE,R=3)                          (compile output of UPDATE)
FILE,TAPE31,BT=I.                           (optional, only if LCAP2
REQUEST(TAPE31,*PF)                          operator STORE is to be used)
RETURN(OLDPL)
ATTACH(LCAPLIB,8LCAP2LIBX,ID=9487)          (attach LCAP2 library)
ATTACH(PLOTLIB,3FTNPLOTLIB)                 (attach plot library)
LIBRARY(LCAPLIB,PLOTLIB)
LDSET(PRESET=ZERO)
LGO.                                        (load and execute LCAP2 program)
CATALOG(TAPE31,8filename,ID=.....,ST=PF6)   (optional, use only if LCAP2
                                             STORE operator was used)
HARDCPY,ST=IBMD8.                           (omit argument if A3 plotter desired)
*EOR                                        (end of record)
                .
                .
  (UPDATE input deck as described
   in previous section)
                .
                .
*EOR                                        (end of record)
------------------------------------------------------------------------------
```

The second form is:

```
--------------------------------------------------------------------------------
                .
(    control cards for accounting   )
                .
FILE,TAPE30,RT=I.                           (optional, use only if old data
ATTACH(TAPE30,lfn,ID=.....,ST=PF6)           is to be restored)
ATTACH(X,8LCAP2CC,ID=9487)                  (attach LCAP2 control card PROC)
BEGIN,LCAP2CC,X.                            (execute PROC)
CATALOG(TAPE31,8filename,ID=.....,ST=PF6)   (optional, use only if LCAP2
                                             STORE operator was used)
HARDCPY,ST=IBMD8.                           (omit argument if A3 plotter desired)
*EOR                                        (end of record)
              .
              .
  (UPDATE input deck as described
   in previous section)
              .
              .
*EOR                                        (end of record)
--------------------------------------------------------------------------------
```

In the second form, the file X will generate the same control cards as the first form except for the (1) FILE,TAPE30,...., (2) ATTACH(TAPE30,...., (3) CATALOG(TAPE31,.... , and (4) HARDCPY. statements. It is recommended that the second form be used unless the user must change some of the control cards. An example when this is necessary is if the print limit is exceeded. The statement LGO. should be changed to LGO(PL=.....) where the value of PL is the number of print lines.

## 4.0 INTERACTIVE LCAP2

Unlike the batch version where the user creates a main FORTRAN program for each job, the interactive version is compiled once and saved as a binary file to be used by all users. Use of this binary file by the user is more complex than simply attaching this file and executing it since some pre and post-processing may be required.

If hardcopy (high resolution electrostatic) plots are to be produced, a plot file PLOT must be created and cataloged by Interactive LCAP2 on the CDC 835 computer. Since the HARDCPY program, which processes the PLOT file, resides on the CDC 176 computer, a separate batch job must be sent to this computer to execute the HARDCPY program. A PROC (procedure) is used to automate this post-processing task so that it will be transparent to the user. PROCs are also used to automate attaching and cataloging of LCAP2 data files. The following subsections describe the creation of the INTERACTIVE LCAP2 binary code and procedures.

### 4.1 JOB STRUCTURE FOR CREATION OF INTERACTIVE LCAP2 BINARY CODE

The basic operations in creating the binary code for Interactive LCAP2 are:

(1) Creation of FORTRAN main program

(2) Compilation of the source code for (1)

(3) Segment loading of routines from LCAP2 and system libraries

(4) Cataloging of binary program created in (2) and (3)

Like the batch version of the program, the UPDATE program is utilized to facilitate the development of the FORTRAN main program. The same UPDATE COMDECK LCAP2, with minor modifications, is used to create the first part of the main program. The input (card images) for the UPDATE program is given by:

```
*IDENT XYZ
*/    CHANGE DEFAULT BUFFER LENGTH FOR FILES
*DELETE LCAP2.3,LCAP2.4
     +,TAPE19=200,TAPE30=461,TAPE31=461,TAPE83=200
     +,TAPE84=461,TAPE85=461,TAPE86=461,TAPE87=461,TAPE89=461
*/    DELETE REFERENCE TO COMMON BLOCKS MATRIX1 AND MDET1
*DELETE EL11163.2,EL11163.7
*INSERT START.1
*DECK MAIN
*CALL LCAP2
*CALL ACOM
      CALL ILCAP2
      END
```

In the batch version, code for the first part of the main program was copied from the UPDATE COMDECK LCAP2. For the interactive version a similar process will be used except that modifications must first be made to COMDECK LCAP2. Statements LCAP2.3 and LCAP2.4 are to be changed as indicated. These are continuation statements in the main program declaration statement which declare files to be used by the program. In the batch version, which uses the CDC 176, buffer space is allocated for the declared files in LCM (large core memory). In the interactive version buffer space occupies SCM (small core memory) instead of LCM. Since the default length is 2003 octal words per file and small core memory is to be conserved, the buffer length for these files was reduced. Statements EL11163.2 through EL11163.7, which declares common blocks MATRIX1 and MDET1, are deleted since they are not needed for Interactive LCAP2. The directives *INSERT START.1 through *CALL LCAP2 perform the same operations as described for the batch version. The *CALL ACOM directive will declare common block ACOM. Subroutine ILCAP2 is the executive routine for interactive LCAP2. Description of this routine is given in Section 6.

## 4.2 JOB SUBMITTAL FOR CREATION OF INTERACTIVE LCAP2 BINARY CODE

The operations given in Section 4.1 are executed in the interactive mode of the CDC 835 by the following PROC:

```
.PROC,XXXX,FF=#FILE.
ATTACH(OLDPL,8LCAP2PL835,ID=9487)       (attach LCAP2 program library)
UPDATE(I=FF)
FTN(I=COMPILE,L=OUTPUT)                 (compile output of UPDATE)
MAP(PART)
ATTACH(OLDLIB,8LCAP2LIB835,ID=9487)     (attach LCAP2 library)
ATTACH(PLOTLIB,3FTNPLOTLIB)             (attach plot library)
LIBRARY(OLDLIB,PLOTLIB)
RFL(120000)                             (request field length)
SEGLOAD(I=FF)                           (segment loader)
LGO.
REVERT.
*EOR                                    (end of record)
              .
              .
  (UPDATE input deck as described
   in Section 4.1)
              .
              .
*EOR                                    (end of record)
              .
              .
  (SEGLOAD directives
   see Appendix B)
              .
              .
*EOR                                    (end of record)
```

To execute the PROC the following SENATOR and INTERCOM commands can be used:

In SENATOR,

```
(1) SAVE TEMP.DATA                    (save above PROC file)
(2) END                               (return to INTERCOM mode)
```

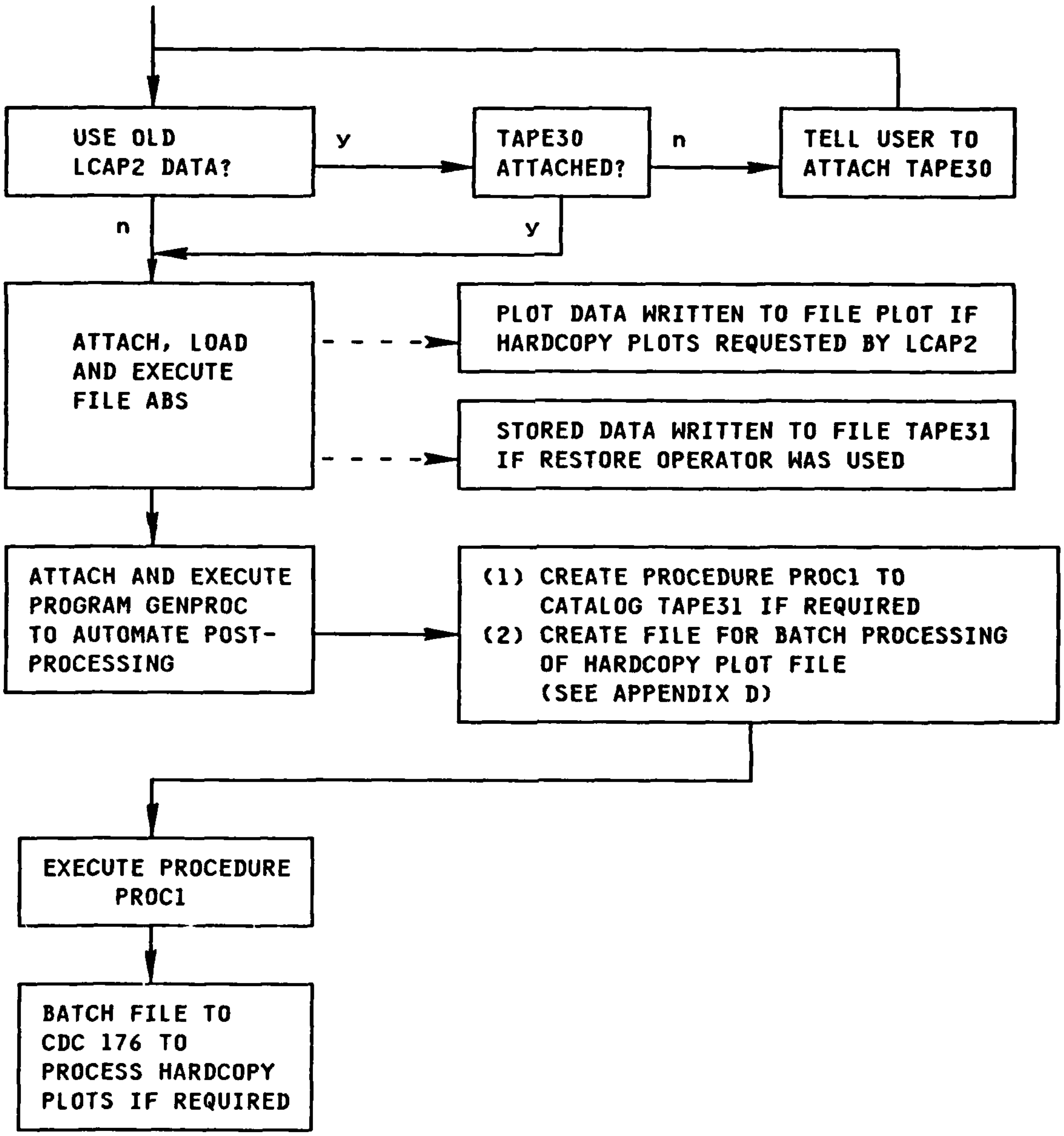
In INTERCOM,

```
(3) REQUEST(ABS,*PF)
(4) ETL 300                           (increase time limit)
(5) BEGIN,XXXX,TEMP                   (execute PROC, file ABS will be
                                       the output of SEGLOAD)
(6) REWIND ABS
(7) CATALOG(ABS,8ABSINTLCAP2,ID=9487,PW=.....)
```

The file ABS created above can be attached and executed by the user if no pre and post-processing is required. Generally though, pre and post-processing will be required. A PROC has been written to perform these operations so that the whole process will be transparent to the user. This is described in the following section.

## 4.3 PROC FOR LOADING AND EXECUTING INTERACTIVE LCAP2

The PROC INTLCAP2 describing the loading and execution of Interactive LCAP2 is given below. The complexity of this process is the result of (1) hardcopy plots must be processed off-line in the batch mode using the CDC 176, (2) there is no higher level interactive command language available, and (3) the desire to simplify user prompts as much as possible.




Block Diagram for PROC INTLCAP2

The top part of the block diagram describes the pre-processing to check if the user has attached the file to be used for restoring previously stored LCAP2 data. The middle part of the block diagram describe the use of the Interactive LCAP2 program. The output blocks from the dotted lines describe data that must be processed after the program exits from ABS.

In order to automate the process to perform the post-processing to catalog TAPE31 and to process the hardcopy PLOT file, the program GENPROC is first executed. The program will first interrogate a file written by Interactive LCAP2 to see if hardcopy plots are to be produced and if TAPE31 is to be cataloged. The user is then prompted for information required for completing this post-processing operation. A procedure PROC1 is then written for cataloging TAPE31. Next, a batch file will be created for processing the hardcopy plots. The control cards, which contain accounting information, are automatically written for the user using system routines to pick this information from data initially logged in by the user.

After GENPROC has been completed, the procedure PROC1, which was just written by GENPROC, will be executed. The batch file created by GENPROC will then be batched from the CDC 835 to the CDC 176.

The code for the procedure INTLCAP2 is given in Appendix C.

The code for the program GENPROC is given in Appendix D.

## 5.0 LCAP2 DATA FORMAT

To simplify program development, fixed size arrays are used by LCAP2 to represent polynomials, transfer functions and matrices. Polynomials up to degree 49 and transfer functions up to degree 49 over 49 can be accommodated. The matrices used for transfer function evaluation by Cramer's method can be as large as 30 x 30.

## 5.1 POLYNOMIAL FORMAT

Polynomials in LCAP2 can be represented in coefficient or in root form. A polynomial in x is represented in coefficient form by

$$\sum_{i=0}^{n} a_i x^i$$

A real array of dimension 51 is used to represent this data. The degree n of this polynomial is stored in the first element of the array. The coefficients $a_i$ of the polynomial are stored in ascending order starting with the second element of the array.

A polynomial in x is represented in root form by

$$a_u x^u \prod_{i=1}^{n} \left( \frac{x}{-alpha_i} + 1 \right)$$

where n is the number of roots, $alpha_i$ is a complex root, u is the number of roots at the origin and $a_u$ is the low order non-zero coefficient. A complex array of dimension 50 is used to represent this data. The number of roots, n, is stored in the real part of the first element of the array. The low order non-zero coefficient, $a_u$, is stored in the imaginary part of the first element of the array. The complex roots are stored in successive elements starting with the second element of the array.

To input coefficient data using the PLDC operator, the array POLY is used to enter coefficient data. To input root data using the PLDR operator, the complex array ROOT is used to enter the root data. For typical use these are the only polynomial FORTRAN arrays which the user needs to be concerned with. All other polynomials used will be referenced with indices as arguments of the LCAP2 operators.

The polynomials which can be referenced with indices as arguments of LCAP2 operators are designated as POLY1, POLY2, POLY3, etc. The first five POLYi's are stored in COMMON/SCMBLK/. All additional POLYi's where i is larger than five are stored on file TAPE84. The format used to represent this data is the same as that described above.

## 5.2 TRANSFER FUNCTION FORMAT

Transfer functions in LCAP2 can be represented in coefficient or in root form. A transfer function in x is represented in coefficient form by

$$\frac{\sum_{i=0}^{n} a_i x^i}{\sum_{j=0}^{m} b_j x^j}$$

A real array of dimension 102 is used to represent this data. The degree n of the numerator is stored in the first element of the array. The coefficients $a_i$ of the numerator are stored in ascending order starting with the second element of the array. The degree m of the denominator is stored in the fifty-second element of the array. The coefficients $b_j$ of the denominator are stored in ascending order starting with the fifty-third element of the array.

A transfer function in x is represented in root form by

$$\frac{a_u x^u \prod_{i=1}^{n} \left( \frac{x}{-\mathrm{alpha}_i} + 1 \right)}{b_v x^v \prod_{j=1}^{m} \left( \frac{x}{-\mathrm{beta}_j} + 1 \right)}$$

where m is the number of numerator roots, $alpha_i$ is a numerator root, u is the number of numerator roots at the origin, $a_u$ is the low order non-zero coefficient of the numerator, m is the number of denominator roots, $beta_j$ is a denominator root, v is the number of denominator roots at the origin, and $b_v$ is the low order non-zero coefficient of the denominator. A complex array of dimension 100 is used to represent this data. The number of numerator roots, n, is stored in the real part of the first element of this array. The low order non-zero coefficient, $a_u$, of the numerator is stored in the imaginary part of the first first element of this array. The complex numerator roots, $alpha_i$ (i=1,n), are stored in successive elements starting with the second element of this array. The number of denominator roots, m, is stored in the real part of the fifty-first element of this array. The low order non-zero coefficient, $b_v$, of the denominator is stored in the imaginary part of the fifty-first element of this array. The complex denominator roots, $beta_j$ (j=1,m), are stored in successive elements starting with the fifty-second element of this array.

To input transfer function data using the SPLDC, ZPLDC or WPLDC operators, corresponding to s, z or w plane loading, the arrays POLYN and POLYD are used to enter the coefficient data for the numerator and denominator, respectively. To input transfer function data using the SPLDR, ZPLDR or WPLDR operators, corresponding to s, z or w plane loading, the complex arrays ROOTN and ROOTD are used to enter root data for the numerator and denominator, respectively. For typical use, these are the only transfer function FORTRAN arrays which the user will need to be concerned with. All other transfer functions used will be referenced with indices as arguments of the LCAP2 operators.

The s plane transfer functions which can be referenced with indices as arguments of LCAP2 operators are designated as SPTF1, SPTF2, SPTF3, etc. The first five SPTFi's are stored in COMMON/SCMBLK/. All additional SPTFi's where i is larger than five are stored on file TAPE85. Corresponding to the s plane, z and w plane transfer functions are designated as ZPTF1, ZPTF2, ZPTF3, etc. and WPTF1, WPTF2, WPTF3, etc., respectively. The first five ZPTFi's and first five WPTFi's are also stored in COMMON/SCMBLK/. All additional ZPTFi's are stored on file TAPE86 and all additional WPTFi's are store on file TAPE87.

## 5.3 MATRIX FORMAT

Cramer's method for transfer function evaluation is given in Example 11 of Ref. 1 and in Example 12 of Ref. 2. The matrix $\underline{M}$(s) can have polynomial elements up to degree four. To input this data into matrices $\underline{M0}$, $\underline{M1}$, $\underline{M2}$, $\underline{M3}$ and $\underline{M4}$ are defined by

$$\underline{M}(s) = \underline{M4}\, s^4 + \underline{M3}\, s^3 + \underline{M2}\, s^2 + \underline{M1}\, s^1 + \underline{M0}$$

A real array of dimension 30 x 30 is used to represent each of these input matrices. The user's dimension of these matrices is specified by the parameter MXM. The highest order of the polynomial elements is specified by the parameter MDEG.

Although a matrix of dimension 30 x 30 can be entered with up to fourth order polynomial elements, there is a restriction that the determinant of this matrix, as computed by the operators DTERM or DETRM, must yield a polynomial of degree less than fifty. This restriction is due to the fixed size polynomial arrays used to save this determinant polynomial.

## 5.4 B VECTOR FORMAT

The original determinant operator, DTERM, required the user to manually change column elements of the matrix in order to obtain the numerator polynomial of a transfer function via Cramer's method. The new version of the determinant operator, DTERM, automatically substitutes the forcing vector into a desired column of the matrix $\underline{M}(s)$. This forcing vector is $\underline{B}(s)$ which can have polynomial elements up to degree four. To input this data input vectors $\underline{B0}$, $\underline{B1}$, $\underline{B2}$, $\underline{B3}$ and $\underline{B4}$ are defined by

$$\underline{B}(s) = \underline{B4}\, s^4 + \underline{B3}\, s^3 + \underline{B2}\, s^2 + \underline{B1}\, s^1 + \underline{B0}$$

A real array of dimension 30 is used to represent each of these input vectors. The user's dimension of these input vectors is specfied by the parameter MXM. The highest order of the polynomial elements is specified by the parameter MDEG. polynomial.

## 6.0 SUBROUTINES

Description of each routine in the LCAP2 subroutine library is given in this section.

### ADDP

#### Identification

SUBROUTINE ADDP - Add Polynomials (Coefficient Form)
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

#### Purpose

Add coefficients of two polynomials

#### Usage

CALL ADDP(A,B,C)

A input - Polynomial coefficient array (LCAP2 format)
B input - Polynomial coefficient array (LCAP2 format)
C output - Polynomial coefficient array (LCAP2 format)

1. EPAD1 (preset=1.E10) in COMMON/HEADDB/ is used to test for negligible coefficients.

#### Method

Coefficients of polynomials A and B are added and stored in polynomial C. A test is then made to see if the highest order coefficient is smaller than all the other coefficients by 1/EPAD1. If it is, then it is considered to be negligible and is set to zero and the order of the polynomial reduced by one. This test is then repeated.

#### Restrictions

The degree of the polynomials must be less than 50.

#### Requirements

COMMON blocks: HEADDB
LCAP2 routines: PCHEK,PEQUAL,PZERO

# AUXM1

## Identification

SUBROUTINE AUXM1 - Auxiliary Subroutine Used With MULE For Computing Determinant Of A Complex Matrix
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Compute determinant of a matrix with complex elements. When MROOT1 is called, AUXM1 is used by subroutine MULE to compute the eigenvalues of the determinant polynomial of a matrix. This subroutine is to be used for the batch version only.

## Usage

```
CALL AUXM1(RTX,FRTX,SC)
```

RTX input - Complex root iterant supplied by MULE
FRTX output - Complex determinant of matrix computed by CXMTX1
SC - not used

1. Matrix data and the computed complex determinant, DET, are stored in COMMON/MDET1/. See description for MROOT1.

2. RTMAX (preset=1.E7) in COMMON/HEADDB/ is used to determine when to terminate iterative procedure by MULE. When MAX(RTX) exceeds RTMAX, all eigenvalues should have been found. FRTX is then set to zero so that MULE will terminate searching for the eigenvalues. If roots to be found are very large, RTMAX should be increased so that it is at least 1.E4 larger than the largest known root.

## Method

Determinant is computed by CXMTX1. See description for MROOT1.

## Restrictions

Maximum dimension of matrix is 30 x 30. Highest order polynomial element is four.

Use for batch version of LCAP2 only. If used for interactive version, program length will be increased because blank common will not be shared by matrix and plot data.

## Requirements

COMMON blocks: HEADDB,MATRIX1,MDET1
LCAP2 routines: CXMTX1

# AUXM2

## Identification

SUBROUTINE AUXM2 - Auxiliary Subroutine Used With MULE For Computing Determinant Of A Complex Matrix
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Compute determinant of a matrix with complex elements. When MROOT2 is called, AUXM2 is used by subroutine MULE to compute the eigenvalues of the determinant polynomial of a matrix. This subroutine is to be used for the interactive version only.

## Usage

CALL AUXM2(RTX,FRTX,SC)

RTX input - Complex root iterant supplied by MULE
FRTX output - Complex determinant of matrix computed by CXMTX1
SC - not used

1. Matrix data and the computed complex determinant, DET, are stored in blank common //. See description for MROOT2.

2. RTMAX (preset=1.E7) in COMMON/HEADDB/ is used to determine when to terminate iterative procedure by MULE. When max(RTX) exceeds RTMAX, all eigenvalues should have been found. FRTX is then set to zero so that MULE will terminate searching for the eigenvalues. If roots to be found are very large, RTMAX should be increased so that it is at least 1.E4 larger than the largest known root.

3. AUXM2 is similar to AUXM1 except that blank common // is used instead of COMMON/MATRIX1/ and COMMON/MDET1/.

## Method

Determinant is computed by CXMTX1. See description for MROOT2.

## Restrictions

Maximum dimension of matrix is 30 x 30. The degree of the polynomial elements must not be greater than 4.

Use for interactive version of LCAP2 only. If used for the batch version, the user must keep track of what data is in blank common if frequency response and determinant operations are intermixed.

<u>Requirements</u>

COMMON blocks: HEADDB,//
LCAP2 routines: CXMTX1

## AUXP

### Identification

SUBROUTINE AUXP - Auxiliary Subroutine Used With MULE For Computing Roots Of A Polynomial
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Evaluate a polynomial in coefficient form. When subroutine PROOT is called, AUXP is used by MULE to compute the roots of a polynomial.

### Usage

```
CALL AUXP(S,FS,SC)
```

| | | |
|---|---|---|
| S | input | - Complex root iterant supplied by MULE |
| FS | output | - Complex value of polynomial evaluated at S |
| SC | | - not used |

1. Coefficients of the polynomial are in polynomial array POLYC (LCAP2 format) of COMMON/CMPOLY/. (They were copied from the arguments of subroutine PROOT into the array POLYC.)

### Method

The polynomial is evaluated in double precision.

### Restrictions

The degree of the polynomial must be less than 50.

### Requirements

COMMON blocks: CMPOLY
LCAP2 routines: none

# AUXP1

## Identification

SUBROUTINE AUXP1 - Evaluate Polynomial (in coefficient form) With A Complex Argument
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Evaluate a polynomial, in coefficient form, for a given complex value of the independent variable.

## Usage

```
CALL AUXP1(POLY,S,FS)
```

| | | |
|---|---|---|
| POLY | input | - Polynomial coefficient array (LCAP2 format) |
| S | input | - Complex value of independent variable |
| FS | output | - Complex value of polynomial evaluated at S |

## Method

The polynomial is evaluated in double precision.

## Restrictions

The degree of the polynomial must be less than 50.

## Requirements

COMMON blocks: none
LCAP2 routines: none

# AUXRT1

## Identification

SUBROUTINE AUXRT1 - Evaluate Polynomial (in root form) With A Complex Argument
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Evaluate a polynomial, in root form, for a given complex value of the independent variable.

## Usage

```
CALL AUXRT1(ROOT,S,FS)
```

| | | |
|---|---|---|
| ROOT | input | - Complex polynomial root array (LCAP2 format) |
| S | input | - Complex value of the independent variable |
| FS | output | - Complex value of polynomial evaluated at S |

## Restrictions

The degree of the polynomial must be less than 50.

## Requirements

COMMON blocks: none
LCAP2 routines: XTRACT

## AXXMRN

### Identification

SUBROUTINE AXXMRN - Evaluate Function Used To Compute Numerator Of Multirate Transform
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Evaluate function used to compute numerator of multirate transform by Sklansky's frequency decomposition method. When subroutine MRXFM is called, AXXMRN is used by subroutine MULE to compute the numerator roots.

### Usage

```
CALL AXXMRN(X,FX,SC)
```

| | | |
|---|---|---|
| X | input | - Complex root iterant supplied by MULE |
| FX | output | - Complex value of function evaluated with X |
| SC | | - not used |

1. Function to be evaluated is defined in subroutine MRXFM and placed in COMMON/COMAXX/. See description for MRXFM.

### Method

(to be documented later)

### Restrictions

The degree of the polynomial must be less than 50.

### Requirements

COMMON blocks: COMAXX,ITEST,HEADDB,TEMPRT
LCAP2 routines: EJKN

## BILNWZ

### Identification

SUBROUTINE BILNWZ - Bilinear Transformation of Polynomial Coefficients From W To Z Plane
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Transform coefficients of w plane polynomial to z plane polynomial using the bilinear transformation.

### Usage

CALL BILNWZ(PIN,POUT)

PIN input - W plane polynomial coefficient array (LCAP2 format)
POUT output - Z plane polynomial coefficient array (LCAP2 format)

### Method

Algorithm by A. C. Davies. (IEEE Trans. On Circuits and Systems, pp 792-794, Nov. 1974)

### Restrictions

This method is not very accurate for higher order polynomials since it utilizes only the coefficients of the polynomial. See description for WZXFM which is more accurate since the roots of the polynomials are used.

### Requirements

COMMON blocks: none
LCAP2 routines: none

# BILNZW

## Identification

SUBROUTINE BILNZW - Bilinear Transformation of Polynomial Coefficients From Z To W Plane
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Transform coefficients of z plane polynomial to w plane polynomial using the bilinear transformation.

## Usage

```
CALL BILNZW(PIN,POUT)
```

PIN input - Z plane polynomial coefficient array (LCAP2 format)
POUT output - W plane polynomial coefficient array (LCAP2 format)

## Method

Algorithm by A. C. Davies. (IEEE Trans. On Circuits and Systems, pp 792-794, Nov. 1974)

The code for this is routine is in subroutine BILNWZ.

## Restrictions

This method is not very accurate for higher order polynomials since it utilizes only the coefficients of the polynomial. See description for ZWXFM which is more accurate since the roots of the polynomials are used.

## Requirements

COMMON blocks: none
LCAP2 routines: none

# BPRINT1

## Identification

SUBROUTINE BPRINT1 - Print Out B Vector
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Print out vectors B0, B1, B2, B3, and B4 which are used in evaluating a transfer function by Cramer's method. BPRINT1 is to be used only for the batch version of LCAP2.

## Usage

CALL BPRINT1

1. B vector data is in COMMON/MATRIX1/. They are described below:

| Parameter | Description |
|---|---|
| MATDIM | DIMENSION of vectors B0,B1,B2,B3,B3,B4 |
| MXM | Dimension of vectors (1-30) |
| MDEG | Highest degree of polynomial element (0-4) |
| B0 | Vector for coefficients of s**0 |
| B1 | Vector for coefficients of s**1 |
| B2 | Vector for coefficients of s**2 |
| B3 | Vector for coefficients of s**3 |
| B4 | Vector for coefficients of s**4 |

## Method

Only the non-zero elements of the vectors are printed out.

## Requirements

COMMON blocks: MATRIX1,PRNCTL
LCAP2 routines: none

## BPRINT2

### Identification

SUBROUTINE BPRINT2 - Print Out B Vector
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Print out vectors B0, B1, B2, B3, and B4 which are used in evaluating a transfer function by Cramer's method. BPRINT2 is to be used only for the interactive version of LCAP2.

### Usage

CALL BPRINT2

1. B vector data is in blank common //. They are described below:

| Parameter | Description |
|---|---|
| MATDIM | DIMENSION of vectors B0,B1,B2,B3,B4 |
| MXM | Dimension of vectors (1-30) |
| MDEG | Highest degree of polynomial element (0-4) |
| B0 | Vector for coefficients of s**0 |
| B1 | Vector for coefficients of s**1 |
| B2 | Vector for coefficients of s**2 |
| B3 | Vector for coefficients of s**3 |
| B4 | Vector for coefficients of s**4 |

### Method

Only the non-zero elements of the matrices are printed out.

### Requirements

COMMON blocks: PRNCTL,//
LCAP2 routines: none

## CNGCO

### Identification

SUBROUTINE CNGCO - Interactive Input Routine To Change Transfer Function Coefficients
CDC FORTRAN 4
E. A. Lee.
Aerospace Corporation

### Purpose

Prompts user for data to change coefficients of an existing transfer function.

### Usage

CALL CNGCO(INDX)

INDX input - Index of transfer function to be changed

1. Transfer functions will be stored in the s, w or z plane as determined by the flag PLN in COMMON/ACOM/. Set PLN=1HS, 1HW or 1HZ for s, w or z plane, respectively, before calling CNGCO.

### Method

Program will print out the transfer function coefficients and ask the user if the data is correct. If not, the user can (1) change the degree or (2) change the coefficients of either or both the numerator or the denominator.

The code for this routine is in subroutine COEFF.

### Restrictions

The degree of the transfer function must be less than 50.

### Requirements

COMMON blocks: ACOM,HEADDB,TFTEMP
LCAP2 routines: FETTFX,PEQUAL,STRTFX

# CNGCOP

## Identification

SUBROUTINE CNGCOP - Interactive Input Routine To Change Polynomial Coefficients
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Prompts user for data to change coefficients of an existing polynomial.

## Usage

```
CALL CNGCOP(INDX)
```

INDX input - Index of polynomial to be changed

## Method

Program will print out the polynomial coefficients and ask the user if the data is correct. If not, the user can (1) change the degree or (2) change the coefficients.

The code for this routine is in subroutine COEFP.

## Restrictions

The degree of the polynomial must be less than 50.

## Requirements

COMMON blocks: ACOM,HEADDB
LCAP2 routines: FETPY,STRPY

# CNGRT

## Identification

SUBROUTINE CNGRT - Interactive Input Routine To Change Transfer Function Roots
CDC FORTRAN 4
E. A. Lee.
Aerospace Corporation

## Purpose

Prompts user for data to change roots of an existing transfer function.

## Usage

CALL CNGRT(INDX)

INDX input - Index of transfer function to be changed

1. Transfer functions will be stored in the s, w or z plane as determined by the flag PLN in COMMON/ACOM/. Set PLN=1HS, 1HW or 1HZ for s, w or z plane, respectively, before calling CNGRT.

## Method

Program will print out the transfer function roots and ask the user if the data is correct. If not, the user can (1) add root values, (2) delete root values, (3) change root values or (4) change the gain value.

The code for this routine is in subroutine IROOT.

## Restrictions

The degree of the transfer function must be less than 50.

## Requirements

COMMON blocks: ACOM,HEADDB,TFTEMP,//
LCAP2 routines: FCNW1,FCNW2,FETTFX,PROOT,PSYNTH,RTCMNT,RTEQU,
STRTFX,XTRACT

## CNGRTP

### Identification

SUBROUTINE CNGRTP - Interactive Input Routine To Change Polynomial Root
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Prompts user for data to change roots of an existing polynomial.

### Usage

```
CALL CNGCOP(INDX)
```

INDX input - Index of polynomial to be changed

### Method

Program will print out the polynomial roots and ask the user if the data is correct. If not, the user can (1) add root values, (2) delete root values, (3) change root values or (4) change the gain value.

The code for this routine is in subroutine IROOTP

### Restrictions

The degree of the polynomial must be less than 50.

### Requirements

COMMON blocks: ACOM,HEADDB,TFTEMP,//
LCAP2 routines: FETPY,PROOT,PSYNTH,RTCMNT,STRPY

# COEFF

## Identification

SUBROUTINE COEFF - Interactive Input Routine For Transfer Function Data In Coefficient Form
CDC FORTRAN 4
F. P. Fernandez and E. A. Lee.
Aerospace Corporation

## Purpose

Prompts user for data to load in transfer function coefficients.

## Usage

CALL COEFF(INDX)

INDX output - Index used to store transfer function which the user entered in response to a prompt

1. Transfer functions will be stored in the s, w or z plane as determined by the flag PLN in COMMON/ACOM/. Set PLN=1HS, 1HW or 1HZ for s, w or z plane, respectively, before calling COEFF.

## Method

Program will prompt the user for transfer function coefficients. After data entry, the program will print out the transfer function and ask the user if the data is correct. If not, the user can (1) change the degree or (2) change the coefficients of either or both the numerator or the denominator. The program then prompts the user for the number where this transfer function is to be stored. This number is returned to the calling program to be used, if necessary, for further processing of LCAP2 operators.

## Restrictions

The degree of the transfer function must be less than 50.

## Requirements

COMMON blocks: ACOM,HEADDB,TFTEMP
LCAP2 routines: FCNW1,FETTFX,PEQUAL,STRTFX

## COEFP

### Identification

SUBROUTINE COEFP - Interactive Input Routine For Polynomial Data In Coefficient Form
CDC FORTRAN 4
F. P. Fernandez and E. A. Lee
Aerospace Corporation

### Purpose

Prompts user for data to load in polynomial coefficients.

### Usage

CALL COEFP(INDX)

INDX output - Index used to store polynomial which the user entered in response to a prompt

### Method

Program will prompt the user for polynomial coefficients. After data entry, the program will print out the polynomial and ask the user if the data is correct. If not, the user can (1) change the degree or (2) change the coefficients. The program then prompts the user for the number where this polynomial is to be stored. This number is returned to the calling program to be used, if necessary, for further processing of LCAP2 operators.

### Restrictions

The degree of the polynomial must be less than 50.

### Requirements

COMMON blocks: ACOM,HEADDB
LCAP2 routines: FETPY,STRPY

# CPPRN

## Identification

SUBROUTINE CPPRN - CP (Central Processing) Time Print Out
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Print out CP time used from beginning of job. This allows the user to determine how much computing time is required for various operations.

## Usage

CALL CPPRN

1. Flag INTFLG (preset=0) in COMMON/INTCOM/, if .NE.0, will suppress the printout. The printout is intended to be used for the batch version of LCAP2.

## Method

Calls system routine SECOND(T) and prints out T.

## Requirements

COMMON blocks: INTCOM
LCAP2 routines: none

## CPYPS

### Identification

SUBROUTINE CPYPS - LCAP2 Operator, Copy Polynomials Into S Plane Transfer Function
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Copy polynomials into an s plane transfer functions using LCAP2 indices. For transfer function evaluation by Cramer's method, this operator is used to define a transfer function after two polynomial determinants have been computed with the use of the DETRM operator.

### Usage

CALL CPYPS(I,J,K)

I input - Index of s plane transfer function where results are to be stored
J input - Index of poly. to be used to define numerator of SPTFi, i=I
K input - Index of poly. to be used to define denominator of SPTFi, i=I

### Method

Copies coefficients of polynomials into a transfer function. If the roots of the polynomials are also defined, these roots are also copied into the transfer function.

### Restrictions

The degree of the polynomials must be less than 50.

### Requirements

COMMON blocks: PRNFLG,TFTEMP
LCAP2 routines: ENDLINE,FETPY,OPPRN,PEQUAL,PYPRN1,RTEQU,STRTFX,TFPRN4

## CPYPW

### Identification

SUBROUTINE CPYPW - LCAP2 Operator, Copy Polynomials Into W Plane Transfer Function
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

This operator is similar to CPYPS except that it is for a w plane transfer function instead of a s plane transfer function.

The code for this routine is in subroutine CPYPS.

## CPYPZ

### Identification

SUBROUTINE CPYPZ - LCAP2 Operator, Copy Polynomials Into Z Plane Transfer Function
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

This operator is similar to CPYPS except that it is for a z plane transfer function instead of a s plane transfer function.

The code for this routine is in subroutine CPYPS.

# CPYSP

## Identification

SUBROUTINE CPYSP - LCAP2 Operator, Copy S Plane Transfer Function Into Polynomials
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Copy s plane transfer function into polynomials using LCAP2 indices.

## Usage

```
CALL CPYSP(I,J,K)
```

```
I  input  - Index of s plane transfer function to be used in copying
J  input  - Index of polynomial equated with the numerator of SPTFi, i=I
K  input  - Index of polynomial equated with the denominator of SPTFi, i=I
```

## Method

Copies coefficients of transfer function into polynomials. If the roots of the transfer function are available, the roots are also stored in the polynomials.

## Restrictions

The degree of the polynomials must be less than 50.

## Requirements

COMMON blocks: PRNFLG,TFTEMP
LCAP2 routines: ENDLINE,FETTFX,OPPRN,PEQUAL,PYPRN4,RTEQU,STRTFX,TFPRN1

## CPYWP

### Identification

SUBROUTINE CPYWP - LCAP2 Operator, Copy W Plane Transfer Function Into Polynomials
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

This operator is similar to CPYSP except that it is for the w plane transfer function instead of the s plane transfer function.

The code for the routine is in subroutine CPYSP.

## CPYZP

### Identification

SUBROUTINE CPYZP - LCAP2 Operator, Copy Z Plane Transfer Function Into Polynomials
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

This operator is similar to CPYSP except that it is for the z plane transfer function instead of the s plane transfer function.

The code for this routine is in subroutine CPYSP.

# CRELIM

## Identification

SUBROUTINE CRELIM - Common Root Elimination
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Eliminate common roots between two complex root arrays in LCAP2 format.

## Usage

```
CALL CRELIM(ROOTN,ROOTD,XROOTN,XROOTD,IQCOM,IQR,IQZ,IPCOM,IPR,IPZ,COMROT)
```

| | | |
|---|---|---|
| ROOTN | input | - Complex polynomial root array (LCAP2 format) for numerator |
| ROOTD | input | - Complex polynomial root array (LCAP2 format) for denominator |
| XROOTN | output | - Complex polynomial root array (LCAP2 format) for numerator |
| XROOTD | output | - Complex polynomial root array (LCAP2 format) for denominator |
| IQCOM | output | - Number of complex roots of numerator |
| IQR | output | - Number of non-zero real roots for numerator |
| IQZ | output | - Number of roots at zero for numerator |
| IPCOM | output | - Number of complex roots for denominator |
| IPR | output | - Number of non-zero real roots for denominator |
| IPZ | output | - Number of roots at zero for denominator |
| COMROT | output | - Complex polynomial root array (LCAP2 format) for the common roots |

1. CRELIM tolerance parameters, ECRE1 (preset=2.E-4) and ECRE2 (preset=1.E-8) are in COMMON/HEADDB/.

2. Diagnostic flag, PRN1, (preset=0) in COMMON/HEADDB/, if .NE.0, will provide additional printout for check out.

## Method

See description of SELCR.

## Restrictions

The degree of the polynomial must be less than 50.

## Requirements

COMMON blocks: HEADDB
LCAP2 routines: RCLAS,RTPRN0

# CXMTX1

## Identification

SUBROUTINE CXMTX1 - Determinant Of A Complex Matrix
CDC COMPASS Assembly Language
H. J. Wertz
Aerospace Corporation

## Purpose

Compute determinant of a matrix with complex elements.

## Usage

```
CALL CXMTX1(A,NVAR,DET,NADIM)
```

```
A      input  - Complex matrix (DIMENSION(NADIM,i), where i .GE. NVAR)
NVAR   input  - Rank of the matrix
DET    output - Complex value of determinant of the matrix
NADIM  input  - Row dimension of A
```

## Method

This subroutine is written in CDC COMPASS assembly language for faster execution time. A FORTRAN version of this routine is also available.

## Requirements

COMMON blocks: none
LCAP2 routines: none

# DAYPRN

## Identification

SUBROUTINE DAYPRN - Date Print Out
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Print out date of execution.

## Usage

CALL DAYPRN

## Method

Calls system routine DATE(A) and prints out the date.

## Requirements

COMMON blocks: none
LCAP2 routines: none

# DETRM

## Identification

SUBROUTINE DETRM - LCAP2 Operator, Determinant Of Matrix With Polynomial Elements (Old Version)

CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Compute polynomial determinant of a matrix with polynomial elements defined by:

$$\underline{M}(s) = \underline{M4}\, s^4 + \underline{M3}\, s^3 + \underline{M2}\, s^2 + \underline{M1}\, s^1 + \underline{M0}$$

The determinant is found by solving for its roots directly and then computing its coefficients.

This subroutine is for the batch version of LCAP2. For interactive LCAP2, see subroutine IDETRM.

## Usage

CALL DETRM(I)

I input - Index where polynomial determinant is to be stored

1. Before DETRM is used, the matrix parameters must first be initialized by calling MINITO (only once).

2. Matrix parameters are in COMMON/MATRIX1/. They are to be set before MROOT1 is called. These parameters are defined below:

| Parameters | Preset | Description |
|---|---|---|
| MXM | 1 | Dimension of matrices (1-30) |
| MDEG | 0 | Highest degree of polynomial element (0-4) |
| M0 | 0 | Matrix for coefficients of s**0 |
| M1 | 0 | Matrix for coefficients of s**1 |
| M2 | 0 | Matrix for coefficients of s**2 |
| M3 | 0 | Matrix for coefficients of s**3 |
| M4 | 0 | Matrix for coefficients of s**4 |

3. When this operator is used for transfer function evaluation via Cramer's method, the user must manually change the appropriate elements of the matrix to account for the forcing vector. An improved version of this operator, DTERM, will perform the substitution of the forcing vector automat-

ically. Use DTERM instead of DETRM. The old version of the determinant operator will be maintained for compatability with old deck setups.

Method

See description for MROOT1.

Restrictions

The dimension of the matrix must not be greater than 30 x 30. The polynomial elements of the matrix must be of degree 4 or less. The degree of the computed polynomial determinant must be less than 50.

Requirements

COMMON blocks: INTCOM,PRNCTL,TFTEMP
LCAP2 routines: HOLLI,MPRINT1,MROOT1,OPMESG,PPRN1,RTPRN2,STRPY

# DTERM

## Identification

SUBROUTINE DTERM - LCAP2 Operator, Determinant Of Matrix With Polynomial Elements (New Version)

CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Transfer function evaluation by Cramer's method for the system described by

$$\underline{M}(s)\ \underline{X}(s) = \underline{B}(s)u$$

where $\underline{M}(s) = \underline{M4}\ s^4 + \underline{M3}\ s^3 + \underline{M2}\ s^2 + \underline{M1}\ s^1 + \underline{M0}$

$\underline{B}(s) = \underline{B4}\ s^4 + \underline{B3}\ s^3 + \underline{B2}\ s^2 + \underline{B1}\ s^1 + \underline{B0}$

$\underline{M0}$, $\underline{M1}$, $\underline{M2}$, $\underline{M3}$, $\underline{M4}$ are square matrices of dimension MXM
$\underline{B0}$, $\underline{B1}$, $\underline{B2}$, $\underline{B3}$, $\underline{B4}$ are vectors of dimension MXM

$\underline{X}(s)$ = State vector of dimension MXM
u = Scalar input

is given by

$$\frac{x_j(s)}{u} = \frac{\det \underline{M}_j(s)}{\det \underline{M}(s)}$$

where $\underline{M}_j(s)$ is equal to $\underline{M}(s)$ with column j replaced by $\underline{B}(s)$.

The operator DTERM will compute the determinant of $\underline{M}_j(s)$. Substitution of $\underline{B}(s)$ into column j will be done automatically by the program.

The determinant is found by solving for its roots directly and then computing its coefficients.

This subroutine is for the batch version of LCAP2. For interactive LCAP2, see subroutine IDTERM.

## Usage

```
CALL DTERM(I,J)
```

I input - Index where polynomial determinant is to be stored
J input - Column where B(s) is to be substituted into
(J=0 interpreted to mean no column substitution)

1. Before DTERM is used, the matrix parameters must first be initialized by calling MINITO (only once).

2. Matrix parameters are in COMMON/MATRIX1/. They are to be set before MROOT1 is called. These parameters are defined below:

| Parameters | Preset | Description |
|---|---|---|
| MXM | 1 | Dimension of matrices and vectors (1-30) |
| MDEG | 0 | Highest degree of polynomial element (0-4) |
| M0 | 0 | Matrix for coefficients of s**0 |
| M1 | 0 | Matrix for coefficients of s**1 |
| M2 | 0 | Matrix for coefficients of s**2 |
| M3 | 0 | Matrix for coefficients of s**3 |
| M4 | 0 | Matrix for coefficients of s**4 |
| B0 | 0 | Vector for coefficients of s**0 |
| B1 | 0 | Vector for coefficients of s**1 |
| B2 | 0 | Vector for coefficients of s**2 |
| B3 | 0 | Vector for coefficients of s**3 |
| B4 | 0 | Vector for coefficients of s**4 |

## Method

If j is not zero, B(s) is substituted into column j of M(s). Subroutine MROOT1 is then called to compute the determinant. Column j of M(s) is then restored to its original value.

## Restrictions

The dimension of the matrix must not be greater than 30 x 30. The polynomial elements of the matrix must be of degree 4 or less. The degree of the computed polynomial determinant must be less than 50.

## Requirements

COMMON blocks: INTCOM,MATRIX1,PRNCTL,TFTEMP
LCAP2 routines: BPRINT1,HOLLI,LEXIT,MPRINT1,MROOT1,OPMESG,PPRN1,RTPRN2,
STRPY

## DOTLINE

### Identification

SUBROUTINE DOTLINE - Print Out One Row Of Dots
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Print out one row of dots to be used for delimiting operations.

### Usage

```
CALL DOTLINE
```

1. Flag PRNFLG3 (preset=1) in COMMON/PRNCTL/, if .EQ.0, will suppress the printout.

### Requirements

COMMON blocks: PRNCTL
LCAP2 routines: none

# EJKN

## Identification

COMPLEX FUNCTION EJKN - COMPUTE e**(j(2pi*k/n))
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Compute exponent of j(2.*pi*k/n) used in evaluating frequency decomposition operations. Used by subroutine AXXMRN when subroutine MRXFM is called to compute the multirate z transform by Sklansky's frequency decomposition method.

## Usage

```
CALL EJKN(k)
```

1. All other arguments and results are in COMMON/COMAXX/. See description for MRXFM.

## Method

For improved computational efficiency, a table look up in COMMON/COMAXX/ is used if k is .LE. 10. This table is created in subroutine MRXFM by calling EJKNI for k=1,..,n. If k is larger than 10, the complex exponent is computed for each call instead of using a table look up. The table look up can be extended past 10 elements if user needs warrant it.

## Requirements

COMMON blocks: COMAXX
LCAP2 routines: none

# EJKNI

## Identification

COMPLEX FUNCTION EJKNI - Initialization of Complex Function EJKN
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Initialization of complex function EJKN(k). Look up table is created and stored in COMMON/COMAXX/. See description for EJKN.

## Usage

```
CALL EJKNI(k)
```

1. Argument k is defined in description of EJKN.

## Method

See description for EJKN.

Code for this routine is in subroutine EJKN.

## Requirements

COMMON blocks: COMAXX
LCAP2 routines: none

## ELPLOT1

### Identification

SUBROUTINE ELPLOT1 - Plot Routines Utilizing Aerospace Routines
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Plot routine for producing higher resolution hardcopy plots.

### Usage

```
CALL ELPLOT1(XVAR,YVAR,CVAR,XTITLE,YTITLE,ATITLE,INFLG)
```

```
XVAR   input  - array of the independent variable
YVAR   input  - array of the dependent variable
CVAR   input  - array of input quantities
                CVAR(1)  - plot segment
                           0 = complete plot   1 = first call of a
                                                   continuation plot
                           2 = continuation    3 = final call of a
                                                   continuation plot
                CVAR(2)  - Number of values in each array
                CVAR(3)  - Starting point for x axis of plot
                CVAR(4)  - Max. acceptable value along the x axis
                           If CVAR(3)=CVAR(4), auto.scaling for x axis
                CVAR(5)  - Units per inch along the x axis
                CVAR(6)  - Constant added to all x values before plotting
                CVAR(7)  - Constant all x values are multiplied by before plotting
                CVAR(8)  - Start point for y axis of plot x axis
                CVAR(9)  - Units per inch along the y axis
                           If =0, auto. scaling for 10 inch y axis
                CVAR(10) - Constant added to all y values before plotting
                CVAR(11) - Constant all y values are multiplied by before plotting
                CVAR(12) - Max. x distance (.01 inch) without lifting pen
                CVAR(13) - Max. y distance (.01 inch) without lifting pen
                CVAR(14) - Intensity of line (2 to 28), nominal is 16
                CVAR(15) - Intensity of zero line
                CVAR(16) - .EQ. 0 for linear plot, .NE.0 for semi-log plot
                CVAR(17) - Number of cycles for semi-log plots
                           0 = automatic, 2 = two cycles, 3 = three cycles
                CVAR(18) - Grid pattern for 835 film plotter
                           =4HNICO for Nichols plot grid
                           =3HSQR for 10x10 grid
                           =6HZPOLAR for polar plot
                CVAR(19) - Number of additional lines of annotation, (0-3)
                CVAR(20) - y position for additional annotation
```

CVAR(21) - =0 for point plot
=1 for x, with numeric labeling, =-1 without
=2 for o, with numeric labeling, =-2 without
CVAR(22) - Numeric for labeling when CVAR(21) .GT. 0
CVAR(23) - =0 for line plot
=1 for step plot

## Method

This subroutine utilizes the Aerospace plot routines. This subroutine was written before the availability of the portable graphics routines such as CAL-COMP and GCS.

## Requirements

COMMON blocks: INTCOM
LCAP2 routines: CPPRN,HOLLI,OSCALE

# ENDLINE

## Identification

SUBROUTINE ENDLINE - Print Out One Row Of Dashes
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Print out one row of dashes and the CP (Central Processing) time. This enables the user to determine the computer time required for various operations.

## Usage

```
CALL ENDLINE
```

1. Flag PRNFLG3 (preset=1) in COMMON/PRNCTL/, if .EQ.0, will suppress the printout.

2. Interactive flag INTFLG (preset=0) in COMMON/INTCOM/, if .NE.0, will suppress print out of CP time. (INTFLG set to 1 for interactive LCAP2)

## Requirements

COMMON blocks: INTCOM,PRNCTL
LCAP2 routines: none

<u>EVLRRT</u>

## Identification

COMPLEX FUNCTION EVLRRT - Evaluate Transfer Funtion In Root Form For A Given Complex Value

CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Evaluate transfer function in root form for a given complex value.

## Usage

```
EVLRRT(ROOT,S)
```

```
ROOT   input  - Complex transfer function root array (LCAP2 format)
S      input  - Complex value of the independent variable
EVLRRT output - Complex value of transfer function evaluated at S
```

## Method

The code for the routine is in subroutine EVLRT.

## Restrictions

The degree of the transfer function must be less than 50.

## Requirements

COMMON blocks: none
LCAP2 routines: XTRACT

## EVLRT

### Identification

COMPLEX FUNCTION EVLRT - Evaluate Polynomial In Root Form For A Given Complex Value

CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Evaluate polynomial in root form for a given complex value.

### Usage

```
EVLRT(ROOT,S)
```

```
ROOT  input  - Complex polynomial root array (LCAP2 format)
S     input  - Complex value of the independent variable
EVLRT output - Complex value of polynomial evaluated at S
```

### Restrictions

The degree of the polynomial must be less than 50.

### Requirements

COMMON blocks: none
LCAP2 routines: XTRACT

## EVMRRT

### Identification

COMPLEX FUNCTION EVMRRT - Evaluate Transfer Function Using Frequency Decomposition
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Evaluate multirate transfer function in root form for a given complex value. The multirate transfer function is defined by Sklansky's frequency decomposition method.

### Usage

```
EVMRRT(IPLANE,TFR,NRATIO,XN)
```

| | | |
|---|---|---|
| IPLANE | input | - =1 for z plane, =-1 for w plane |
| TFR | input | - Transfer function root array (LCAP2 format) at the faster sampling rate |
| NRATIO | input | - Ratio of output/input sampling periods |
| XN | input | - Complex frequency at the faster input sampler used to evaluate the frequency response |
| EVMRRT | output | - Complex value of transfer function response evaluated at XN |

### Method

Subroutine EVLRRT is used to evaluate the transfer functions in Sklansky's frequency decomposition method.

### Restrictions

The degree of the transfer function must be less than 50.

### Requirements

COMMON blocks: none
LCAP2 routines: EVLRRT

# FAUXW

## Identification

COMPLEX FUNCTION FAUXW - Evaluate Response Of User-Supplied W Plane Transfer Function

CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Evaluate response of user-supplied w plane transfer function. The w plane function is defined by a user-supplied subroutine.

## Usage

FAUXW(CFUNC)

CFUNC input - Name of user-supplied subroutine. Must be declared with an EXTERNAL statement in the calling program.
FAUXW output - Complex value of the response

1. Independent w plane frequency used in evaluation of the response is computed by the program using real frequency U in COMMON/FRQBLK/ and sampling period SAMPT in COMMON/HEADDB/.

## Requirements

COMMON blocks: FRQBLK,HEADDB
LCAP2 routines: none

## FAUXWM

### Identification

COMPLEX FUNCTION FAUXWM - Evaluate Multirate Response Of User-Supplied
W Plane Transfer Function

CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Evaluate multirate response, by Sklansky's frequency decomposition method, of user-supplied w plane transfer function. The w plane transfer function is defined by a user-supplied subroutine.

### Usage

FAUXWM(CFUNC,M,T)

| | | |
|---|---|---|
| CFUNC | input | - Name of user-supplied subroutine. Must be declared with an EXTERNAL statement in the calling program. |
| M | input | - Ratio of output/input sampling periods |
| T | input | - Sampling period of slower output sampler |
| FAUXWM | output | - Complex value of the response |

1. Independent w plane variable used in evaluation of the response is computed by the program using real frequency U in COMMON/FRQBLK/.

### Requirements

COMMON blocks: FRQBLK,HEADDB
LCAP2 routines: none

# FAUXZ

## Identification

COMPLEX FUNCTION FAUXZ - Evaluate Response Of User-Supplied Z Plane Transfer Function

CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Evaluate response of user-supplied z plane transfer function. The z plane function is defined by a user-supplied subroutine.

## Usage

FAUXZ(CFUNC)

CFUNC input - Name of user-supplied subroutine. Must be declared with an EXTERNAL statement in the calling program.
FAUXZ output - Complex value of the response

1. Independent z plane frequency used in evaluation of the response is computed by the program using real frequency U in COMMON/FRQBLK/ and sampling period SAMPT in COMMON/HEADDB/.

## Requirements

COMMON blocks: FRQBLK,HEADDB
LCAP2 routines: none

## FAUXZM

### Identification

COMPLEX FUNCTION FAUXZM - Evaluate Multirate Response Of User-Supplied Z Plane Transfer Function
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Evaluate multirate response, by Sklansky's frequency decomposition method, of user-supplied z plane transfer function. The z plane transfer function is defined by a user-supplied subroutine.

### Usage

FAUXZM(CFUNC,M,T)

| | | |
|---|---|---|
| CFUNC | input | - Name of user-supplied subroutine. Must be declared with an EXTERNAL statement in the calling program. |
| M | input | - Ratio of output/input sampling periods |
| T | input | - Sampling period of slower output sampler |
| FAUXZM | output | - Complex value of the response |

1. Independent z plane variable used in evaluation of the response is computed by the program using real frequency U in COMMON/FRQBLK/.

### Requirements

COMMON blocks: FRQBLK,HEADDB
LCAP2 routines: none

# FCNPLN

## Identification

FUNCTION FCNPLN - Compute Hollerith Word For IPLANE Flag
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Compute Hollerith word for IPLANE flag word.

## Usage

```
FCNPLN(IPLANE)

IPLANE  input  - =0, -1, 1
FCNPLN  output - = 1HS if IPLANE = 0
                 = 1HW if IPLANE = -1
                 = 1HZ if IPLANE = 1
```

## Requirements

COMMON blocks: none
LCAP2 routines: none

## FCNW1

### Identification

FUNCTION FCNW1 - Hollerith Representation of Transfer Function (Coefficient
Form) Identifier
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Compute Hollerith representation of transfer function (coefficient form) identifier.

### Usage

```
FCNW1(IPLANE,I)

IPLANE  input  - =0 for s plane, =-1 for w plane, =1 for z plane
I       input  - Integer between 0 and 100
FCNW1   output - =6HSPTFi, i = I, for IPLANE = 0
                 =6HWPTFi, i = I, for IPLANE = -1
                 =6HZPTFi, i = I, for IPLANE = 1
```

### Requirements

COMMON blocks: none
LCAP2 routines: HOLLI

## FCNW2

### Identification

FUNCTION FCNW2 - Hollerith Representation of Transfer Function (Root Form) Identifier
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Compute Hollerith representation of transfer function (root form) identifier.

### Usage

```
     FCNW2(IPLANE,I)

IPLANE  input  - =0 for s plane, =-1 for w plane, =1 for z plane
I       input  - Integer between 0 and 100
FCNW1   output - =6HROOTi, i = I, for IPLANE = 0
                 =6HROOTi, i = I, for IPLANE = -1
                 =6HROOTi, i = I, for IPLANE = 1
```

### Method

The code for this function is in FCNW1.

### Requirements

COMMON blocks: none
LCAP2 routines: HOLLI

# FETPY

## Identification

SUBROUTINE FETPY - Fetch Polynomial
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Fetch polynomial in LCAP2 format.

## Usage

```
CALL FETPY(INDX,TFPOLY,TFROOT,IN)
```

```
INDX    input  - Index of polynomial to be fetched
TFPOLY  output - Polynomial coefficient array (LCAP2 format)
TFROOT  output - Complex polynomial root array (LCAP2 format), only if IN=0
IN      output - = 1 when only coefficients are available
                 = 0 when both coefficients and roots are available
```

1. If INDX .GT.5, polynomial data must have been previously saved using STRPY.

2. NPYCNT in COMMON/TFPCNT/ is the number of polynomial records on sequential file TAPE84.

## Method

If INDX is 1,2,.. or 5, the data is read from COMMON/SCMBLK/. If INDX is greater than 5, the polynomial data is read from the sequential file TAPE84.

## Restrictions

For INDX greater than 5, the index must have been previously used in a call to STRPY. In the batch program, if INDX has not been defined yet, the program will abort. In the interactive program, if INDX has has not been defined yet, the program will suspend the current LCAP2 operator and reenter (bypassing the normal return) subroutine ILCAP2 at entry IRECOV.

## Requirements

COMMON blocks: HEADDB,INTCOM,SCMBLK,TFPCNT
LCAP2 routines: IRECOV

# FETSTF

## Identification

SUBROUTINE FETSTF - Fetch S Plane Transfer Function
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

To fetch s plane transfer function in LCAP2 format.

## Usage

```
CALL FETSTF(INDX,TFPOLY,TFROOT,IN,ID)

INDX    input  - Index of s plane transfer function to be fetched
TFPOLY  output - Transfer function coefficient array (LCAP2 format)
TFROOT  output - Complex transfer function root array (LCAP2 format),
                 only if IN and ID equal 0.
IN      output - = 1 when only numerator coefficients are available
                 = 0 when both numerator coefficients and roots are available
ID      output - = 1 when only denominator coefficients are available
                 = 0 when both denominator coefficients and roots are available
```

1. If INDX .GT.5 transfer function data must have been previously saved using STRSTF or STRTFX.

2. NSPCNT in COMMON/TFPCNT/ is the number of s plane transfer function records on the sequential file TAPE85.

## Method

If INDX is 1,2,.. or 5, the data is read from COMMON/SCMBLK/. If INDX is greater than 5, the transfer function data is read from the sequential file TAPE85.

## Restrictions

For INDX greater than 5, the index must have been previously used in a call to STRSTF or STFTFX. In the batch program, if INDX has not been defined yet, the program will abort. In the interactive program, if INDX has has not been defined yet, the program will suspend the current LCAP2 operator and reenter (bypassing the normal return) subroutine ILCAP2 at entry IRECOV.

## Requirements

COMMON blocks: HEADDB,INTCOM,SCMBLK,TFPCNT
LCAP2 routines: FCNW1,IRECOV

# FETTFX

## Identification

SUBROUTINE FETTFX - Fetch Transfer Function
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Fetch transfer function in LCAP2 format. Similar to FETSTF, FETWTF and FETZTF except the identifier for plane is included as an argument.

## Usage

```
CALL FETTFX (IPLANE,INDX,TFPOLY,TFROOT,IN,ID)

IPLANE  input  - =0 for s, =-1 for w, =1 for z plane
INDX    input  - Index of transfer function to be fetched
TFPOLY  output - Transfer function coefficient array (LCAP2 format)
TFROOT  output - Complex transfer function root array (LCAP2 format),
                 only if IN and ID equal 0.
IN      output - = 1 when only numerator coefficients are available
                 = 0 when both numerator coefficients and roots are available
ID      output - = 1 when only denominator coefficients are available
                 = 0 when both denominator coefficients and roots are available
```

## Method

This subroutine calls either FETSTF, FETWTF or FETZTF. See description for FETSTF.

## Requirements

COMMON blocks: HEADDB,SCMBLK,TFPCNT
LCAP2 routines: FETSTF,FETWTF,FETZTF

## FETWTF

### Identification

SUBROUTINE FETWTF - Fetch W Plane Transfer Function
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

To fetch w plane transfer function in LCAP2 format.

### Usage

CALL FETWTF(INDX,TFPOLY,TFROOT,IN,ID)

| | | |
|---|---|---|
| INDX | input | - Index of w plane transfer function to be fetched |
| TFPOLY | output | - Transfer function coefficient array (LCAP2 format) |
| TFROOT | output | - Complex transfer function root array (LCAP2 format), only if IN and ID equal 0. |
| IN | output | - = 1 when only numerator coefficients are available<br>= 0 when both numerator coefficients and roots are available |
| ID | output | - = 1 when only denominator coefficients are available<br>= 0 when both denominator coefficients and roots are available |

1. If INDX .GT.5 transfer function data must have been previously saved using STRWTF or STRTFX.

2. NWPCNT in COMMON/TFPCNT/ is the number of w plane transfer function records on the sequential file TAPE86.

### Method

If INDX is 1,2,.. or 5, the data is read from COMMON/SCMBLK/. If INDX is greater than 5, the transfer function data is read from the sequential file TAPE86.

The code for this routine is in subroutine FETSTF.

### Restrictions

For INDX greater than 5, the index must have been previously used in a call to STRWTF or STFTFX. In the batch program, if INDX has not been defined yet, the program will abort. In the interactive program, if INDX has has not been defined yet, the program will suspend the current LCAP2 operator and reenter (bypassing the normal return) subroutine ILCAP2 at entry IRECOV.

Requirements

COMMON blocks: HEADDB,INTCOM,SCMBLK,TFPCNT
LCAP2 routines: FCNW1,IRECOV

## FETZTF

### Identification

SUBROUTINE FETZTF - Fetch Z Plane Transfer Function
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

To fetch z plane transfer function in LCAP2 format.

### Usage

CALL FETZTF(INDX,TFPOLY,TFROOT,IN,ID)

```
INDX    input  - Index of z plane transfer function to be fetched
TFPOLY  output - Transfer function coefficient array (LCAP2 format)
TFROOT  output - Complex transfer function root array (LCAP2 format),
                 only if IN and ID equal 0.
IN      output - = 1 when only numerator coefficients are available
                 = 0 when both numerator coefficients and roots are available
ID      output - = 1 when only denominator coefficients are available
                 = 0 when both denominator coefficients and roots are available
```

1. If INDX .GT.5 transfer function data must have been previously saved using STRZTF or STRTFX.

2. NZPCNT in COMMON/TFPCNT/ is the number of z plane transfer function records on the sequential file TAPE87.

### Method

If INDX is 1,2,.. or 5, the data is read from COMMON/SCMBLK/. If INDX is greater than 5, the transfer function data is read from the sequential file TAPE87.

The code for this routine is in subroutine FETSTF.

### Restrictions

For INDX greater than 5, the index must have been previously used in a call to STRZTF or STFTFX. In the batch program, if INDX has not been defined yet, the program will abort. In the interactive program, if INDX has has not been defined yet, the program will suspend the current LCAP2 operator and reenter (bypassing the normal return) subroutine ILCAP2 at entry IRECOV.

Requirements

COMMON blocks: HEADDB,INTCOM,SCMBLK,TFPCNT
LCAP2 routines: FCNW1,IRECOV

## FPLOT

### Identification

SUBROUTINE FPLOT - Frequency Response Plotting Routine
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Compute frequency response plots such as the Bode, Nichols and Nyquist plots.

### Usage

```
CALL FPLOT(NI,DEGPT,DBPT,OMEGPT,CXR,CXI,FDBMX)
```

| | | |
|---|---|---|
| NI | input | - Number of plot points |
| DEGPT | input | - Array of degree points |
| DBPT | input | - Array of DB points |
| OMEGPT | input | - Array of omega points |
| CXR | input | - Array of real part of response |
| CXI | input | - Array of imaginary part of response |
| FDBMX | input | - Maximum DB value of array DBPT |

1. The plot options are determined by the variables in COMMON/HEADDB/. See description for SFREQ.

2. Additional annotation on the right portion of the Nichols plot can be created with a user-supplied subroutine USRNOTE.

### Method

This subroutine is a driver to ELPLOT1.

### Restrictions

The number of plot points must not be greater than 1500.

### Requirements

COMMON blocks: AWORDS,HEADDB,PLOT1
LCAP2 routines: DAYPRN,ELPLOT1,GRAF1,OSCALE,USRNOTE

## FPLOT1

### Identification

SUBROUTINE FPLOT1 - Frequency Response Plotting Routine
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Compute frequency response plots such as the Bode, Nichols and Nyquist plots. Includes code for prompting by interactive LCAP2.

### Usage

```
CALL FPLOT1(NI,DEGPT,DBPT,OMEGPT,CXR,CXI,FDBMX,IREDO)
```

| | | |
|---|---|---|
| NI | input | - Number of plot points |
| DEGPT | input | - Array of degree points |
| DBPT | input | - Array of DB points |
| OMEGPT | input | - Array of omega points |
| CXR | input | - Array of real part of response |
| CXI | input | - Array of imaginary part of response |
| FDBMX | input | - Maximum DB value of array DBPT |
| IREDO | output | - (For interactive LCAP2 only) .NE.0 if user wants to redefine frequency range from calling program. This flag can be tested by the calling program so that FPLOT1 can be called again. |

1. The plot options are determined by the variables in COMMON/HEADDB/. See description for SFREQ.

2. NPLOTS in COMMON/PLOT1/ is the number of hardcopy plots created. This variable is used at the end of a job to determine how many hardcopy records have been generated.

3. Interactive flag INTFLG(preset=0) of COMMON/INTFLG/ must be set .NE.0 for interactive LCAP2.

4. Additional annotation on the right portion of the Nichols plot can be created with a user-supplied subroutine USRNOTE.

### Method

This subroutine is a driver to ELPLOT1.

### Restrictions

The number of plot points must not be greater than 1500.

Requirements

COMMON blocks: AWORDS,INTCOM,HEADDB,PLOT1
LCAP2 routines: ELPLOT1,GRAF1,ITITLE,OSCALE,USRNOTE

# FREQS

## Identification

SUBROUTINE FREQS - LCAP2 Operator, Frequency Response Using A
User-Supplied Function

CDC FORTRAN 4

E. A. Lee

Aerospace Corporation

## Purpose

Compute frequency response of an arbitrary s plane transfer function.

## Usage

```
CALL FREQS(FAUX1)
```

FAUX1 input - Name of user-supplied subroutine. Must be declared with an EXTERNAL statement in the calling program.

1. Frequency response parameters are in COMMON/HEADDB/. See description of SFREQ. The user need not be concerned with passing the arguments for computing the frequency response. It is automatically done by LCAP2.

## Method

This subroutine call FREQS1.

## Restrictions

FREQS is to be used only for the batch version of LCAP2.

## Requirements

COMMON blocks: SCMBLK
LCAP2 routines: FREQS2,FREQW2,FREQZ2

## FREQSI1

### Identification

SUBROUTINE FREQSI1 - Interactive Prompts For Frequency Response Operators
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Interactive code to prompt user for frequency response parameters.

### Usage

```
CALL FREQSI1(IENTRY,MULTI,IPRN1)

IENTRY input  - =1 for s plane
                =2 for z plane
                =3 for w plane
                =4 for multirate z plane
                =5 for multirate w plane
MULTI  input  - .NE.0 for multirate sampling
IPRN1 output  - =0 to suppress tabular printout of response
                .NE.0 to print out response
```

### Method

This subroutine is called by subroutine FREQS3. This code was not placed into FREQS3 so that it can be placed into a higher tree for the segment loader.

### Requirements

COMMON blocks: FRQBLK,HEADDB
LCAP2 routines: none

# FREQS2

## Identification

SUBROUTINE FREQS2 - Frequency Response Of An S Plane Transfer Function In LCAP2 Format

CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Compute frequency response of an s plane transfer function. The transfer function can be specified in the standard LCAP2 format or it can be an arbitrary user-defined transfer function. A user defined transfer function does not have to have the same limitations as the standard LCAP2 format.

## Usage

```
CALL FREQS2(A,FAUX)
```

| | | |
|---|---|---|
| A | input | - Transfer function coefficient array (LCAP2 format) |
| FAUX | input | - Name of complex function to be used for evaluating the transfer function. (Must be declared with an EXTERNAL statement in the calling program). If array A is to be evaluated, enter SFAUX. |

1. Frequency response parameters are in COMMON/HEADDB/. See description of SFREQ. The user need not be concerned with passing the arguments for computing the frequency response. It is automatically done by LCAP2.

2. Computed response variables are stored in blank common //. See description for SFREQ.

## Method

Arguments of this subroutine determine the transfer function to be used. If the second argument is SFAUX, the code in FREQS1 and SFAUX is written so that the array A will be used as the transfer function. If the second argument is not SFAUX, the user must supply his own complex function to evaluate a transfer function. For an example see description for SFAUX1.

## Requirements

COMMON blocks: //
LCAP2 routines: FPLOT1,FREQS3

# FREQS3

## Identification

SUBROUTINE FREQS3 -
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Compute frequency response of a transfer function. This subroutine is called by FREQS2.

## Usage

```
CALL FREQS3(IENTRY,A,FAUX)
```

| | | |
|---|---|---|
| IENTRY | input | - =0 for s plane, =-1 for w plane, =1 for z plane |
| A | input | - Transfer function coefficient array (LCAP2 format) |
| FAUX | input | - Name of complex function to be used for evaluating the transfer function. (Must be declared with an EXTERNAL statement in the calling program). If array A is to be evaluated, enter SFAUX if IENTRY=1, ZFAUX if IENTRY=2 or 4, WFAUX if IENTRY=3 or 5. |

1. Frequency response parameters are in COMMON/HEADDB/. See description of SFREQ. The user need not be concerned with passing the arguments for computing the frequency response. It is automatically done by LCAP2.

2. Computed response variables are stored in blank common //. See description for SFREQ.

## Method

## Requirements

COMMON blocks: FRQBLK,HEADDB,INTCOM,//
LCAP2 routines: FREQSI1

## FREQW

### Identification

SUBROUTINE FREQW - LCAP2 Operator, W Plane Frequency Response Using A User-Supplied Function
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Compute w plane frequency response of an arbitrary w plane transfer function.

### Usage

```
CALL FREQW(FAUX1)
```

FAUX1 input - Name of user-supplied subroutine. Must be declared with an EXTERNAL statement in the calling program.

1. Frequency response parameters are in COMMON/HEADDB/. See description of SFREQ. The user need not be concerned with passing the arguments for computing the frequency response. It is automatically done by LCAP2.

### Method

This subroutine calls FREQW1.

The code for this routine is in subroutine FREQS.

### Restrictions

FREQW is to be used only for the batch version of LCAP2.

### Requirements

COMMON blocks: SCMBLK
LCAP2 routines: FREQS2,FREQW2,FREQZ2

## FREQWM1

### Identification

SUBROUTINE FREQWM1 - Multirate Frequency Response Of A W Plane Transfer Function In LCAP2 Format (Batch Version)
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Compute the multirate (fast input, slow output) frequency response of a w plane transfer function. The transfer function can be specified in the standard LCAP2 format or it can be an arbitrary user-defined transfer function. A user-defined transfer function does not have to have the same limitations as the standard LCAP2 format. FREQWM1 is to be used for the batch version of LCAP2.

### Usage

```
CALL FREQWM1(A,FAUX)
```

A  input - Transfer function coefficient array (LCAP2 format)
FAUX  input - Name of complex function to be used for evaluating the transfer function. (Must be declared with an EXTERNAL statement in the calling program). If array A is to be evaluated, enter WFAUX.

1. The sampling period, SAMPT, of COMMON/HEADDB/ and the integer ratio of the (output/input) sampling periods, MMTGER, of COMMON/FRQBLK/ must be set before FREQWM1 is called.

2. Frequency response parameters are in COMMON/HEADDB/. See description of SFREQ. The user need not be concerned with passing the arguments for computing the frequency response. It is automatically done by LCAP2.

3. Computed response variables are stored in blank common //. See description for SFREQ.

### Method

Sklansky's frequency decomposition method is used to compute the frequency response.

Arguments of this subroutine determine the transfer function to be used. If the second argument is WFAUX, the code in FREQWM1 and WFAUX is written so that the array A will be used as the transfer function. If the second argument is not WFAUX, the user must supply his/her own complex function to evaluate a transfer function. For an example see description for SFAUX1.

The code for this routine is in subroutine FREQS1.

<u>Restrictions</u>

This is to be used only for the batch version of LCAP2.

<u>Requirements</u>

COMMON blocks: FRQBLK,HEADDB,//
LCAP2 routines: FPLOT

# FREQWM2

## Identification

SUBROUTINE FREQWM2 - Multirate Frequency Response Of A W Plane Transfer Function In LCAP2 Format (Interactive Version)
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Compute the multirate (fast input, slow output) frequency response of a w plane transfer function. The transfer function can be specified in the standard LCAP2 format or it can be an arbitrary user-defined transfer function. A user defined transfer function does not have to have the same limitations as the standard LCAP2 format. This is to be used only for interactive version of LCAP2.

## Usage

```
CALL FREQWM2(A,FAUX)
```

A     input  - Transfer function coefficient array (LCAP2 format)
FAUX  input  - Name of complex function to be used for evaluating the transfer function. (Must be declared with an EXTERNAL statement in the calling program). If array A is to be evaluated, enter WFAUX.

1. The sampling period, SAMPT, of COMMON/HEADDB/ and the integer ratio of the (output/input) sampling periods, MMTGER, of COMMON/FRQBLK/ must be set before FREQWM2 is called.

2. Frequency response parameters are in COMMON/HEADDB/. See description of SFREQ. The user need not be concerned with passing the arguments for computing the frequency response. It is automatically done by LCAP2.

3. Computed response variables are stored in blank common //. See description for SFREQ.

## Method

Sklansky's frequency decomposition method is used to compute the frequency response.

Arguments of this subroutine determine the transfer function to be used. If the second argument is WFAUX, the code in FREQWM2 and WFAUX is written so that the array A will be used as the transfer function. If the second argument is not WFAUX, the user must supply his own complex function to evaluate a transfer function. For an example see description for SFAUX1.

To improve segment loading in the interactive version of LCAP2, the code in FREQWM1 was segmented into FREQS2, FREQS3 and FREQSI1. Subroutine FREQS2 is a driver routine to FREQS3. The code for this routine is in subroutine FREQS2.

**Restrictions**

FREQWM2 is to be used only for the interactive version of LCAP2. If used for the batch version a larger memory length will be required for loading.

**Requirements**

COMMON blocks: FRQBLK,HEADDB,//
LCAP2 routines: FPLOT1,FREQS3

# FREQW1

## Identification

SUBROUTINE FREQW1 - Frequency Response Of A W Plane Transfer Function
In LCAP2 Format (Batch Version)
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Compute frequency response of a w plane transfer function. The transfer function can be specified in the standard LCAP2 format or it can be an arbitrary user-defined transfer function. A user-defined transfer function does not have to have the same limitations as the standard LCAP2 format. FREQW1 is to be used for the batch version of LCAP2.

## Usage

```
CALL FREQW1(A,FAUX)
```

A input - Transfer function coefficient array (LCAP2 format)
FAUX input - Name of complex function to be used for evaluating the transfer function. (Must be declared with an EXTERNAL statement in the calling program). If array A is to be evaluated, enter WFAUX.

1. The sampling period, SAMPT of COMMON/HEADDB/ must set before FREQW1 is called.

2. Frequency response parameters are in COMMON/HEADDB/. See description of SFREQ. The user need not be concerned with passing the arguments for computing the fraquency response. It is automatically done by LCAP2.

3. Computed response variables are stored in blank common //. See description for SFREQ.

## Method

Arguments of this subroutine determine the transfer function to be used. If the second argument is WFAUX, the code in FREQW1 and WFAUX is written so that the array A will be used as the transfer function. If the second argument is not WFAUX, the user must supply his own complex function to evaluate a transfer function. For an example see description for SFAUX1.

The code for this routine is in subroutine FREQS1.

Restrictions

FREQW1 is to be used only for the batch version of LCAP2.

Requirements

COMMON blocks: FRQBLK,HEADDB,//
LCAP2 routines: FPLOT

# FREQW2

## Identification

SUBROUTINE FREQW2 - Frequency Response Of A W Plane Transfer Function In LCAP2 Format (Interactive Version)

CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Compute frequency response of a w plane transfer function. The transfer function can be specified in the standard LCAP2 format or it can be an arbitrary user-defined transfer function. A user-defined transfer function does not have to have the same limitations as the standard LCAP2 format. FREQW2 is to be used only for the interactive version of LCAP2.

## Usage

```
CALL FREQW2(A,FAUX)
```

| | | |
|---|---|---|
| A | input | - Transfer function coefficient array (LCAP2 format) |
| FAUX | input | - Name of complex function to be used for evaluating the transfer function. (Must be declared with an EXTERNAL statement in the calling program). If array A is to be evaluated, enter WFAUX. |

1. Frequency response parameters are in COMMON/HEADDB/. See description of SFREQ. The user need not be concerned with passing the arguments for computing the frequency resprnse. It is automatically done by LCAP2.

2. Computed response variables are stored in blank common //. See description for SFREQ.

## Method

Arguments of this subroutine determine the transfer function to be used. If the second argument is WFAUX, the code in FREQW2 and WFAUX is written so that the array A will be used as the transfer function. If the second argument is not WFAUX, the user must supply his own complex function to evaluate a transfer function. For an example see description for SFAUX1.

To improve segment loading in the interactive version of LCAP2, the code in FREQW1 was segmented into FREQS2, FREQS3 and FREQSI1. Subroutine FREQS2 is a driver routine to FREQS3. The code for this routine is in subroutine in FREQS2.

## Restrictions

FREQW2 is to be used only for the interactive version of LCAP2. If used for the batch version a larger memory length will be required for loading.

## Requirements

COMMON blocks: //
LCAP2 routines: FPLOT1

## FREQZ

### Identification

SUBROUTINE FREQZ - LCAP2 Operator, Z Plane Frequency Response Using A User-Supplied Function
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Compute frequency response of an arbitrary z plane transfer function.

### Usage

```
CALL FREQZ(FAUX1)
```

FAUX1 input - Name of user-supplied subroutine. Must be declared with an EXTERNAL statement in the calling program.

1. Frequency response parameters are in COMMON/HEADDB/. See description of SFREQ. The user need not be concerned with passing the arguments for computing the frequency response. It is automatically done by LCAP2.

### Method

This subroutine calls FREQZ1.

The code for this routine is in subroutine FREQS.

### Restrictions

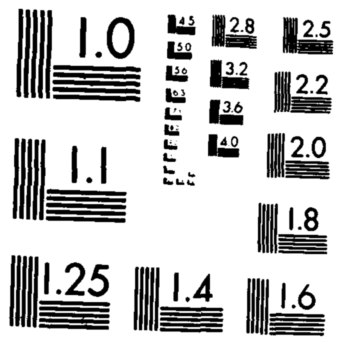
FREQZ is to be used only for the batch version of LCAP2.

### Requirements

COMMON blocks: SCMBLK
LCAP2 routines: FREQS2,FREQW2,FREQZ2

AD-A141 386 LCAP2 (LINEAR CONTROLS ANALYSIS PROGRAM) VOLUME 3 SOURCE CODE DESCRIPTION..(U) AEROSPACE CORP EL SEGUNDO CA GUIDANCE AND CONTROL DIV E A LEE 15 NOV 83 2/4

UNCLASSIFIED TR-0084(9975)-1-VOL-3 SD-TR-84-06-VOL-3 F/G 9/2 NL




MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

# FREQZM1

## Identification

SUBROUTINE FREQZM1 - Multirate Frequency Response Of A Z Plane Transfer Function In LCAP2 Format (Batch Version)
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Compute the multirate (fast input, slow output) frequency response of a z plane transfer function. The transfer function can be specified in the standard LCAP2 format or it can be an arbitrary user-defined transfer function. A user-defined transfer function does not have to have the same limitations as the standard LCAP2 format. FREQZM1 is to be used for the batch version of LCAP2.

## Usage

```
CALL FREQZM1(A,FAUX)
```

| | | |
|---|---|---|
| A | input | - Transfer function coefficient array (LCAP2 format) |
| FAUX | input | - Name of complex function to be used for evaluating the transfer function. (Must be declared with an EXTERNAL statement in the calling program). If array A is to be evaluated, enter ZFAUX. |

1. The sampling period, SAMPT, of COMMON/HEADDB/ and the integer ratio of the (output/input) sampling periods, MMTGER, of COMMON/FRQBLK/ must be set before FREQZM1 is called.

2. Frequency response parameters are in COMMON/HEADDB/. See description of SFREQ. The user need not be concerned with passing the arguments for computing the frequency response. It is automatically done by LCAP2.

3. Computed response variables are stored in blank common //. See description for SFREQ.

## Method

Sklansky's frequency decomposition method is used to compute the frequency response.

Arguments of this subroutine determine the transfer function to be used. If the second argument is ZFAUX, the code in FREQZM1 and ZFAUX is written so that the array A will be used as the transfer function. If the second argument is not ZFAUX, the user must supply his/her own complex function to evaluate a transfer function. For an example see description for SFAUX1.

The code for this routine is in subroutine FREQS1.

**Restrictions**

FREQZM1 is to be used only for the batch version of LCAP2.

**Requirements**

COMMON blocks: FRQBLK,HEADDB,//
LCAP2 routines: FPLOT

# FREQZM2

## Identification

SUBROUTINE FREQZM2 - Multirate Frequency Response OF A Z Plane Transfer Function In LCAP2 Format (Interactive Version)
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Compute the multirate (fast input, slow output) frequency response of a z plane transfer function. The transfer function can be specified in the standard LCAP2 format or it can be an arbitrary user-defined transfer function. A user-defined transfer function does not have to have the same limitations as the standard LCAP2 format. FREQZM2 is to be used for the interactive version of LCAP2.

## Usage

```
CALL FREQZM2(A,FAUX)
```

| | | |
|---|---|---|
| A | input | - Transfer function coefficient array (LCAP2 format) |
| FAUX | input | - Name of complex function to be used for evaluating the transfer function. (Must be declared with an EXTERNAL statement in the calling program). If array A is to be evaluated, enter ZFAUX. |

1. The sampling period, SAMPT, of COMMON/HEADDB/ and the integer ratio of the (output/input) sampling periods, MMTGER, of COMMON/FRQBLK/ must be set before FREQZM2 is called.

2. Frequency response parameters are in COMMON/HEADDB/. See description of SFREQ. The user need not be concerned with passing the arguments for computing the frequency response. It is automatically done by LCAP2.

3. Computed response variables are stored in blank common //. See description for SFREQ.

## Method

Sklansky's frequency decomposition method is used to compute the frequency response.

Arguments of this subroutine determine the transfer function to be used. If the second argument is ZFAUX, the code in FREQZM2 and ZFAUX is written so that the array A will be used as the transfer function. If the second argument is not ZFAUX, the user must supply his/her own complex function to evaluate a transfer function. For an example see description for SFAUX1.

To improve segment loading in the interactive version of LCAP2, the code in FREQZ1 was segmented into FREQS2, FREQS3 and FREQSI1. Subroutine FREQS2 is a driver routine to FREQS3.

The code for this routine is in subroutine FREQS2.

Restrictions

FREQZM2 is to be used only for the batch version of LCAP2. If used for the batch version a larger memory length will be required for loading.

Requirements

COMMON blocks: //
LCAP2 routines: FPLOT1,FREQS3

# FREQZ1

## Identification

SUBROUTINE FREQZ1 - Frequency Response Of A Z Plane Transfer Function
In LCAP2 Format (Batch Version)
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Compute frequency response of a z plane transfer function. The transfer function can be specified in the standard LCAP2 format or it can be an arbitrary user-defined transfer function. A user-defined transfer function does not have to have the same limitations as the standard LCAP2 format. FREQZ1 is to be used for the batch version of LCAP2.

## Usage

```
CALL FREQZ1(A,FAUX)
```

| | | |
|---|---|---|
| A | input | - Transfer function coefficient array (LCAP2 format) |
| FAUX | input | - Name of complex function to be used for evaluating the transfer function. (Must be declared with an EXTERNAL statement in the calling program). If array A is to be evaluated, enter ZFAUX. |

1. The sampling period, SAMPT of COMMON/HEADDB/ must set before FREQZ1 is called.

2. Frequency response parameters are in COMMON/HEADDB/. See description of SFREQ. The user need not be concerned with passing the arguments for computing the frequency response. It is automatically done by LCAP2.

3. Computed response variables are stored in blank common //. See description for SFREQ.

## Method

Arguments of this subroutine determine the transfer function to be used. If the second argument is ZFAUX, the code in FREQZ1 and ZFAUX is written so that the array A will be used as the transfer function. If the second argument is not ZFAUX, the user must supply his/her own complex function to evaluate a transfer function. For an example see description for SFAUX1.

The code for this routine is in subroutine FREQS1.

## Restrictions

FREQZ1 is to be used only for the batch version of LCAP2.

## Requirements

COMMON blocks: FRQBLK,HEADDB,//
LCAP2 routines: FPLOT

# FREQZ2

## Identification

SUBROUTINE FREQZ2 - Frequency Response Of A Z Plane Transfer Function
In LCAP2 Format (Interactive Version)

CDC FORTRAN 4

E. A. Lee

Aerospace Corporation

## Purpose

Compute frequency response of a z plane transfer function. The transfer function can be specified in the standard LCAP2 format or it can be an arbitrary user-defined transfer function. A user-defined transfer function does not have to have the same limitations as the standard LCAP2 format. FREQZ2 is to be used only for the interactive version of LCAP2.

## Usage

```
CALL FREQZ2(A,FAUX)
```

| | | |
|---|---|---|
| A | input | - Transfer function coefficient array (LCAP2 format) |
| FAUX | input | - Name of complex function to be used for evaluating the transfer function. (Must be declared with an EXTERNAL statement in the calling program). If array A is to be evaluated, enter ZFAUX. |

1. Frequency response parameters are in COMMON/HEADDB/. See description of SFREQ. The user need not be concerned with passing the arguments for computing the frequency response. It is automatically done by LCAP2.

2. Computed response variables are stored in blank common //. See description for SFREQ.

## Method

Arguments of this subroutine determine the transfer function to be used. If the second argument is ZFAUX, the code in FREQZ2 and ZFAUX is written so that the array A will be used as the transfer function. If the second argument is not ZFAUX, the user must supply his/her own complex function to evaluate a transfer function. For an example see description for SFAUX1.

To improve segment loading in the interactive version of LCAP2, the code in FREQZ1 was segmented into FREQS2, FREQS3 and FREQSI1. Subroutine FREQS2 is a driver routine to FREQS3.

The code for this routine is in subroutine FREQS2.

## Restrictions

FREQZ2 is to be used only for the interactive version of LCAP2. If used for the batch version a larger memory length will be required for loading.

## Requirements

COMMON blocks: //
LCAP2 routines: FPLOT1

## GRAF1

### Identification

SUBROUTINE GRAF1 - Aerospace Printer Plot Routine
CDC FORTRAN 4
B. Gold, modified by O. Drummond & E. A. Lee
Aerospace Corporation

### Purpose

GRAF1 provides the means to plot one to ten functions on a single full page of printer output paper.

### Usage

```
CALL GRAF1(NP,NFCN,X,F,XMN,XMX,YMN,YMX,NDUM,OPT,KDIM,LOGFLG,ICYCLE)
```

```
NP      input  - Number of plot points
NFCN    input  - Number of functions to be plotted
X       input  - Array of the independent variable
F       input  - Array of the dependent variable
XMN     input  - Minimum x axis
XMX     input  - Maximum x axis
YMN     input  - Minimun y axis
YMX     input  - Maximun y axis
NDUM    input  - Number of columns used for x axis
OPT     input  - .EQ.0 for different range on both axis
                 .EQ.1 for same range on both axis
KDIM    input  - Row dimension of F
LOGFLG  input  - .NE.0 for semi-log plot
ICYCLE  input  - Number of cycles on x axis (1,2 or 3)
```

### Requirements

COMMON blocks: none
LCAP2 routines: none

## HEADINI,(I=1,5)

### Identification

SUBROUTINE HEADINi - Heading Statement For Entering Plot Titles (i=1,5)
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Simple FORTRAN statement for entering Hollerith data into the plot array HEAD of COMMON/HEADDB/ used for labeling plot titles.

### Usage

```
CALL HEADINi(INDX,WORD)
```

INDX input - Pointer to array HEAD of COMMON/HEADDB/ where array WORD will be copied into
WORD input - Hollerith data with format 10H... if i=1
20H... if i=2
. .
50H... if i=5

1. First line of plot title is in HEAD(i),i=1,7
   Second line of plot title is in HEAD(i),i=8,14
   Third line of plot title is in HEAD(i),i=15,21
   Fourth line of plot title is in HEAD(i),i=22,28

2. Example: CALL HEADIN2(8,20HTHIS IS AN EXAMPLE ) will yield,

   HEAD(3)=10HTHIS IS AN
   HEAD(4)=10HEXAMPLE

   which will appear as the second line of the plot title.

3. First line of plot title will appear at the top of the plot. Second line will begin a YANOT units from the bottom of page (full scale defined from 0-10 units). YANOT (preset=9.6) is in COMMON/HEADDB/.

### Requirements

COMMON blocks: HEADDB
LCAP2 routines: none

## HELP

### Identification

SUBROUTINE HELP - Help Subroutine For Interactive LCAP2
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Help subroutine for interactive LCAP2. User is prompted for:

1. General descriptiicn

2. List of polynomial operators

3. List of s plane transfer function operators

4. List of z plane transfer function operators

5. List of w plane transfer function operators

6. List of miscellaneous operators

7. List of polynomial, s,w,z plane and misc. operators

8. Explain use of list directed input for data entry

9. Exit HELP

### Usage

CALL HELP

### Restrictions

Used only by the interactive version of LCAP2.

### Requirements

COMMON blocks: none
LCAP2 routines: none

## HOLLI

### Identification

FUNCTION HOLLI - Converts Integer To Hollerith Characters Left Justified
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Converts integer to Hollerith character left justified.

### Usage

```
     HOLLI(I)

I      input  - Integer between 0 and 100, or else -1 or -2
HOLLI  output - Hollerith representation of I, left justified, if I is
                between 0 and 100.
                = 2HN  if I=-1
                = 2HD  if I=-2
                = Blank otherwise
```

### Requirements

COMMON blocks: none
LCAP2 routines: none

# IARGO

## Identification

SUBROUTINE IARGO - Interactive Driver Routine For Operators With No Input Argument
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Interactive driver routine for LCAP2 operators with no input arguments. LCAP2 operators END, NOP, RESTORE and STORE are processed in this subroutine.

## Usage

```
CALL IARGO(JEXIT)
```

JEXIT output - .EQ.1 if operation END is found
.EQ.0 if operation END is not found

1. This routine is called by ILCAP2.

## Requirements

COMMON blocks: ACOM,INTCOM,PRNCTL
LCAP2 routines: IRSTOR,ISTORE

# IARG1A

## Identification

SUBROUTINE IARG1A - Interactive Driver Routine For Operators With One Input Argument
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Interactive driver routine for some of the LCAP2 operators with one input argument. LCAP2 operators PCNGC, PCNGR, PEQU, PLDC, PLDR, PPRN, PRTS, SELCR, SNORM, SPCNGC, SPCNGR, SPEQU, SPLDC, SPLDR, SPPRN, SPRTS, WELCR, WNORM, WPCNGC, WPCNGR, WPEQU, WPLDC, WPLDR, WPPRN, WPRTS, ZELCR, ZNORM, ZPCNGC, ZPCNGR, ZPEQU, ZPLDC, ZPLDR, ZPPRN, ZPRTS, CPYSP, CPYWP, CPYZP and DETRM are processed in this subroutine.

## Usage

CALL IARG1A

1. This routine is called by ILCAP2.

## Method

Prompting for arguments of the LCAP2 operators (transfer function or polynomial number, or new data) is done in this subroutine. The appropriate subroutine is then called to implement the operator. For the operator PEQU, SPEQU, WPEQU, or ZPEQU, the user is prompted for the number where the resultant polynomial or transfer function is to be stored.

## Requirements

COMMON blocks: ACOM,HEADDB,INTCOM,PRNCTL
LCAP2 routines: CNGCO,CNGCOP,CNGRT,CNGRTP,COEFF,COEFP,CPYSP,CPYPW,CPYZP, ENDLINE,IDETRM,IDTERM,IROOT,IROOTP,PEQU,PPRN,PRTS,SELCR, SNORM,SPEQU,SPPRN,SPRTS,WELCR,WNORM,WPEQU,WPPRN,WPRTS, ZELCR,ZNORM,ZPEQU,ZPPRN,ZPRTS

# IARG1B

## Identification

SUBROUTINE IARG1B - Interactive Driver Routine For Operators With One Input Argument
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Interactive driver routine for some of the LCAP2 operators with one input argument. LCAP2 operators SFREQ, SLOCI, STIME, WFREQ, WMRFQ, WLOCI, ZFREQ, ZMRFQ, ZLOCI, ZTIME are processed in this subroutine.

## Usage

```
CALL IARG1B
```

1. This routine is called by ILCAP2.

## Method

Prompting for arguments of the LCAP2 operators (transfer function or polynomial number, or new data) is done in this subroutine. The appropriate subroutine is then called to implement the operator.

## Requirements

COMMON blocks: ACOM,HEADDB,PRNCTL
LCAP2 routines: COEFF,FREQS,FREQW,FREQZ,IROOT,SFAUX1,SFREQ,SLOCI,STIME,
WFAUX1,WFREQ,WLOCI,WMRFQ,ZFAUX1,ZFREQ,ZLOCI,ZMRFQ,
ZTIME

# IARG2

## Identification

SUBROUTINE IARG2 - Interactive Driver Routine For Operators With Two Input Arguments
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Interactive driver routine for some of the LCAP2 operators with two input argument. LCAP2 operators PADD, PSUB, PMPY, SPADD, SPSUB, SPMPY, SPDIV, WPADD, WPSUB, WPMPY, WPDIV, ZPADD, ZPSUB, ZPMPY, ZPDIV, CPYPS, CPYPW and CPYPZ are processed in this subroutine.

## Usage

CALL IARG2

1. This routine is called by ILCAP2.

## Method

Prompting for arguments of the LCAP2 operators (transfer function or polynomial number, or new data) is done in this subroutine. The appropriate subroutine is then called to implement the operator. The user is then prompted for the number where the polynomial or transfer function is to be stored.

## Requirements

COMMON blocks: ACOM,PRNCTL
LCAP2 routines: COEFF,COEFP,CPYPS,CPYPW,CPYPZ,IROOT,IROOTP,PADD,PMPY,
PSUB,SPADD,SPDIV,SPSUB,WPADD,WPDIV,WPMPY,WPSUB,ZPADD,
ZPDIV,ZPMPY,ZPSUB

# IDETRM

## Identification

SUBROUTINE IDETRM - Interactive Input Of Matrix Data And Calculation Of Determinant
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Interactive input of matrix data and calculation of determinant.

## Usage

CALL IDETRM

1. Before IDETRM is used, the matrix parameters must be initialized by calling MINIT2 (once only).

2. Matrix data is stored in blank common //. Since blank common is also used for storing plot points, the matrix data is also stored in sequential file TAPE83 so that it can later be loaded back into blank common when necessary.

## Method

Upon entry blank common // is either restored with old matrix data from TAPE83 or else it is zeroed out. The program then prompts the user for matrix data. Subroutine MROOT2 is called to compute the determinant polynomial. The user is then prompted for the number where this polynomial is to be stored.

## Restrictions

The dimension of the matrix must not be greater than 30 x 30. The polynomial elements of the matrix must be or degree 4 or less. The degree of the computed polynomial determinant must be less than 50.

## Requirements

COMMON blocks: ACOM,PRNCTL,TFTEMP,//
LCAP2 routines: ENDLINE,HOLLI,MPRINT2,MROOT2,OPPRN,PPRN1,RTPRN2,
STRPY

## IDTERM

### Identification

SUBROUTINE IDTERM - Interactive Input Of Matrix Data And Calculation Of Determinant (New Version)
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Interactive input of matrix data and calculation of determinant with automatic substitution of the forcing vector as described in subroutine DTERM.

### Usage

CALL IDTERM

1. Before IDTERM is used, the matrix parameters must be initialized by calling MINIT2 (once only).

2. Matrix data is stored in blank common //. Since blank common is also used for storing plot points, the matrix data is also stored in sequential file TAPE83 so that it can later be loaded back into blank common when necessary.

### Method

Upon entry blank common // is either restored with old matrix data from TAPE83 or else it is zeroed out. The program then prompts the user for matrix and forcing vector data. Subroutine MROOT2 is called to compute the determinant polynomial. The user is then prompted for the number where this polynomial is to be stored.

### Restrictions

The dimension of the matrix must not be greater than 30 x 30. The polynomial elements of the matrix must be or degree 4 or less. The degree of the computed polynomial determinant must be less than 50.

### Requirements

COMMON blocks: ACOM,PRNCTL,TFTEMP,//
LCAP2 routines: BPRINT2,ENDLINE,HOLLI,MPRINT2,MROOT2,OPPRN,PPRN1,RTPRN2,
STRPY

## ILCAP2

### Identification

SUBROUTINE ILCAP2 - Interactive LCAP2 Executive Subroutine
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Executive subroutine for Interactive LCAP2. This subroutine is to be called by a short main program.

### Usage

CALL ILCAP2

### Method

The first part of this subroutine connects the input and output files to the terminal and initializes the program for interactive use. The program then prompts the user for the LCAP2 operation desired. The operation entered is then read and the appropriate subroutines called to perform the specified operation.

This routine may also be entered at entry IRECOV by a direct call from subroutine FETPY, STRSTF, STRWTF, or STRZTF (bypassing the normal returns) when an undefined argument for an LCAP2 operator is detected.

When the END operator is specified by the user the following operations are performed before the program is terminated. Three words are written to file TAPE89 for post processing of plots and data files by a seperate INTERCOM program. The terminal is then disconnected from the input and output files.

### Requirements

COMMON blocks: ACOM,HEADDB,INTCOM,PLOT1
LCAP2 routines: HELP,IARGO,IARG1A,IARG1B,IARG2,INITO,INTXFM,LEXIT,
MINIT2

# INITO

## Identification

SUBROUTINE INITO - Initialization Of LCAP2 Parameters
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Initialization of LCAP2 parameters.

## Usage

CALL INITO

1. This routine must be called before any LCAP2 subroutines are called.

2. Data in COMMON/SCMBLK/ is initialized in this routine.

3. PRNFLGi flags of COMMON/PRNCTL/ are initialized in this routine.

4. This subroutine should only be called once since it initializes the hard-copy plot counter and the counters for the sequential files used for storing s, w and z plane transfer functions.

## Requirements

COMMON blocks: FRQBLK,HEADDB,LENGTH,OVCOM,PLOT1,PRNCTL,SCMBLK,TFPCNT
LCAP2 routines: none

## INTXFM

### Identification

SUBROUTINE INTXFM - Interactive Driver Routine For The Transform Operators
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Interactive driver routine for LCAP2 transformation operators. LCAP2 operators SWXFM, SZXFM, SWMRX, SZMRX, WZXFM, ZWXFM, ZVCNG, WMRXFM, ZMRXFM, WSXFM, and ZSXFM are processed in this subroutine.

### Usage

CALL INTXFM

1. This routine is called by ILCAP2.

### Method

Prompting for arguments of the LCAP2 operators (transfer function number or new data) is done in this subroutine. The appropriate subroutine is then called to implement the operator. The user is then prompted for the number where the resultant transfer function is to be stored.

### Requirements

COMMON blocks: ACOM,HEADDB,PRNCTL
LCAP2 routines: COEFF,IROOT,SWMRX,SWXFM,SZMRX,SZXFM,WMRXFM,WSXFM,WZXFM,
ZMRXFM,ZSXFM,ZVCNG1,ZWXFM

## IROOT

### Identification

SUBROUTINE IROOT - Interactive Input Routine For Transfer Function Data In Root Form
CDC FORTRAN 4
F. P. Fernandez and E. A. Lee
Aerospace Corporation

### Purpose

Prompt user for data to load in transfer function root data.

### Usage

CALL IROOT(INDX)

INDX output - Index used to store transfer function which the user entered in response to a prompt

1. The transfer function will be stored as an s, w or z plane transfer function as determined by flag PLN of COMMON/ACOM/. Set PLN=1HS, 1HW or 1HZ for s, w or z plane, respectively, before calling this subroutine.

### Method

Program will prompt the user for transfer function root data. After data entry, the program will print out the transfer function and ask the the user if the data is correct. If not, the user can (1) add root values, (2) delete root values, (3) change root values or (4) change the gain value. The program then prompts the user for the number where this transfer function is to be stored. This number is returned to the calling program to be used, if necessary, for further processing of LCAP2 operators.

### Restrictions

The number of roots of the transfer function must be less than 50.

### Requirements

COMMON blocks: ACOM,HEADDB,TFTEMP,//
LCAP2 routines: FCNW1,FCNW2,FETTFX,PROOT,PSYNTH,RTCMNT,RTEQU,STRTFX,
XTRACT

# IROOTP

## Identification

SUBROUTINE IROOTP - Interactive Input Routine For Polynomial Data In Root Form
CDC FORTRAN 4
F. P. Fernandez and E. A. Lee
Aerospace Corporation

## Purpose

Prompt user for data to load in polynomial roots.

## Usage

CALL IROOTP(INDX)

INDX output - Index used to store polynomial which the user entered in response to a prompt

## Method

Program will prompt the user for polynomial data. After data entry, the program will print out the polynomial and ask the user if the data is correct. If not, the user can (1) add root values, (2) delete root values, (3) change root values or (4) change the gain value. The program then prompts the user for the number where this polynomial is to be stored. This number is returned to the calling program to be used, if necessary, for further processing of LCAP2 operators.

## Restrictions

The number of roots of the polynomial must be less than 50.

## Requirements

COMMON blocks: ACOM,HEADDB,TFTEMP,//
LCAP2 routines: FETPY,PROOT,PSYNTH,RTCMNT,STRPY,XTRACT

## IRSTOR

### Identification

SUBROUTINE IRSTOR - Restore Polynomial, Transfer Function And Matrix Data
For Interactive LCAP2
CDC FORTRAN 4
C. L. Wong and E. A. Lee
Aerospace Corporation

### Purpose

Restore data from a previous interactive or batch job for a restart capability in Interactive LCAP2.

### Usage

CALL IRSTOR

### Method

Reads in data stored on TAPE30 and copies it into COMMON/SCMBLK/ and the sequential files TAPE84, TAPE85, TAPE86, TAPE87 and TAPE83. For more details see description for ISTORE.

### Restrictions

File TAPE30 must be attached before executing LCAP2.

### Requirements

COMMON blocks: ACOM,HEADDB,INTCOM,SCMBLK,TFPCNT,//
LCAP2 routines: BPRINT2,ENDLINE,MPRINT2,PYPRN1,TFPRN1

## ISTORE

### Identification

SUBROUTINE ISTORE - Store Polynomial, Transfer Function and Matrix Data
For Interactive LCAP2
CDC FORTRAN 4
C. L. Wong and E. A. Lee
Aerospace Corporation

### Purpose

Store data from an LCAP2 interactive job for a restart capability. This data can be accessed in a subsequent interactive or batch job by using the RESTORE operator.

### Usage

CALL ISTORE

1. The routine will prompt the user for 70 alphanumeric characters to be used to identify the data to be stored. This information will be printed out when this data is restored in a subsequent job.

### Method

Data will be saved on file TAPE31. The first record will be alphanumeric information entered by the user in response to a prompt. The second record will be information from COMMON/TFPCNT/ which describes the number of polynomials and transfer functions saved on various files. The third record will be LCAP2 parameters from HEAD(101) through HEAD(900) of COMMON/HEADDB/.

Polynomials and s, w and z plane transfer functions with LCAP2 indices 1 through 5 are stored in COMMON/SCMBLK/. These polynomials and transfer functions, regardless if they have been used by the user, will be the next data copied onto file TAPE31. Next, polynomials from file TAPE84 will be copied onto file TAPE31. Then s, w and z plane transfer functions from files TAPE85, TAPE86 and TAPE87, respectively, will be copied onto file TAPE31. Finally, matrix data from TAPE83 will be copied onto file TAPE31.

### Requirements

COMMON blocks: ACOM,HEADDB,INTCOM,SCMBLK,TFPCNT,//
LCAP2 routines: BPRINT2,ENDLINE,MPRINT2,PYPRN1,TFPRN1

# ITITLE

## Identification

SUBROUTINE ITITLE - Interactive Input Routine For Entering Title On Plots
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Interactive input routine for entering title on plots. Used by subroutine FPLOT1.

## Usage

CALL ITITLE(HEAD)

HEAD output - Hollerith array (dimension of at least 7 words)

## Method

Terminal will prompt the user for one line of alphanumeric input (70 characters or less) for labeling title of plots. The data will be returned in array HEAD in COMMON/HEADDB/.

## Requirements

COMMON blocks: none
LCAP2 routines: none

## LEXIT

### Identification

SUBROUTINE LEXIT - LCAP2 Exit Routine
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Exit routine with call to plot routine to "clear out" plot buffer before terminating program.

### Usage

CALL LEXIT

### Method

Counter NPLOTS in COMMON/PLOT1/ is checked to see if any records were written to the hardcopy file. If .NE.0, subroutine ENPLOT is called before exiting program with CALL EXIT.

### Requirements

COMMON blocks: PLOT1
LCAP2 routines: none

# MATROT1

## Identification

SUBROUTINE MATROT2 - Compute Determinant Polynomial Of A Matrix
Using A User-Supplied Function
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Compute determinant polynomial of a matrix with polynomial elements using a user-supplied function.

## Usage

```
CALL MATROT1(POLY1,ROOT,AUXSUB,N,NRTS)
```

| | | |
|---|---|---|
| POLY1 | output | - Polynomial coefficient array (LCAP2 array) |
| ROOT | output | - Polynomial root array (LCAP2 array) |
| AUXSUB | input | - Name of user-supplied subroutine. Must be declared with an EXTERNAL statement in the calling program. |
| N | input | - Dimension of matrix |
| NRTS | input | - Max. number of roots to be found |

1. If the user-supplied subroutine is AUXM1, then this subroutine will yield results identical to subroutine MROOT1.

2. User-supplied subroutine can be written so that the dimension of the matrix and the degree of the polynomial elements are different than those of MROOT1.

## Method

Generalized root finding subroutine MULE and the user-supplied subroutine are used to determine the roots of the determinant. As part of the iterative root finding procedure, MULE will pass a root iterant to the user-supplied subroutine which will then evaluate the matrix at this frequency.

## Requirements

COMMON blocks: MDET1,HEADDB,CMPOLY
LCAP2 routines: MULE,PEQUAL,PSYNTH,PZERO,RCLAS,RTPRNO,RZERO

# MINITO

## Identification

SUBROUTINE MINITO - Initialization of LCAP2 Matrix Parameters
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Initialization of LCAP2 matrix parameters used for evaluation of transfer function by Cramer's method. MINITO is to be used only for the batch version of LCAP2.

## Usage

```
CALL MINITO
```

1. This routine must be called before subroutine MROOT1 or DETRM is used.

2. Matrix data is in COMMON/MATRIX1/.

## Method

The matrix data in COMMON/MATRIX1/ which are initialized are:

| Parameter | Initialized Value | Description |
|---|---|---|
| MATDIM | 30 | DIMENSION of square matrices M0,M1,M2,M3,M4 |
| MXM | 1 | Dimension of matrices (1-30) |
| MDEG | 0 | Highest degree of polynomial element (0-4) |
| M0 | 0 | Matrix for coefficients of s**0 |
| M1 | 0 | Matrix for coefficients of s**1 |
| M2 | 0 | Matrix for coefficients of s**2 |
| M3 | 0 | Matrix for coefficients of s**3 |
| M4 | 0 | Matrix for coefficients of s**4 |

Unlike MINIT2 (the interactive version of MINITO), matrix data is not in blank common, which can be over written by plot data which shares blank common. The user does not have to be concerned with the availability of this data once this routine is entered.

## Requirements

COMMON blocks: MATRIX1,MDET1
LCAP2 routines: none

# MINIT2

## Identification

SUBROUTINE MINIT2 - Initialization of LCAP2 Matrix Parameters
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Initialization of LCAP2 matrix parameters used for evaluation of transfer function by Cramer's method. MINIT2 is to be used only for the interactive version of LCAP2.

## Usage

```
CALL MINIT2
```

1. This routine must be called before subroutine MROOT2 or IDETRM is used.

2. Matrix data is in blank common //.

## Method

*To save memory,* blank common // is shared by matrix data and plot data.

The matrix data in blank common which are initialized are:

| Parameter | Initialized Value | Description |
|---|---|---|
| MATDIM | 30 | DIMENSION of square matrices M0,M1,M2,M3,M4 |
| MXM | 1 | Dimension of matrices (1-30) |
| MDEG | 0 | Highest degree of polynomial element (0-4) |
| M0 | 0 | Matrix for coefficients of s**0 |
| M1 | 0 | Matrix for coefficients of s**1 |
| M2 | 0 | Matrix for coefficients of s**2 |
| M3 | 0 | Matrix for coefficients of s**3 |
| M4 | 0 | Matrix for coefficients of s**4 |
| NDIMA | 30 | Dimension of square matrix AMATRIX |
| AMATRIX | 0 | Complex matrix used to evaluate determinant |

After initialization, this data is also saved on sequential file TAPE83 so that blank common can be restored if it is over-written by plot data.

## Requirements

COMMON blocks: //
LCAP2 routines: none

## MPRINT1

### Identification

SUBROUTINE MPRINT1 - Print Out Matrix Data
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Print out matrices M0, M1, M2, M3 and M4 which describe a set of linear transformed equations to be used for evaluating a transfer function by Cramer's method. MPRINT1 is to be used only for the batch version of LCAP2.

### Usage

CALL MPRINT1

1. Matrix data is in COMMON/MATRIX1/. The matrix data are described below:

| Parameter | Description |
|---|---|
| MATDIM | DIMENSION of square matrices M0,M1,M2,M3,M4 |
| MXM | Dimension of matrices (1-30) |
| MDEG | Highest degree of polynomial element (0-4) |
| M0 | Matrix for coefficients of s**0 |
| M1 | Matrix for coefficients of s**1 |
| M2 | Matrix for coefficients of s**2 |
| M3 | Matrix for coefficients of s**3 |
| M4 | Matrix for coefficients of s**4 |

### Method

Only the non-zero elements of the matrices are printed out.

### Requirements

COMMON blocks: MATRIX1,PRNCTL
LCAP2 routines: none

## MPRINT2

### Identification

SUBROUTINE MPRINT2 - Print Out Matrix Data
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Print out matrices M0, M1, M2, M3 and M4 which describe a set of linear transformed equations to be used for evaluating a transfer function by Cramer's method. MPRINT2 is to be used only for the interactive version of LCAP2.

### Usage

CALL MPRINT2

1. Matrix data is in blank common //. Data in this blank common used for the printout are described below:

| Parameter | Description |
|---|---|
| MATDIM | DIMENSION of square matrices M0,M1,M2,M3,M4 |
| MXM | Dimension of matrices (1-30) |
| MDEG | Highest degree of polynomial element (0-4) |
| M0 | Matrix for coefficients of s**0 |
| M1 | Matrix for coefficients of s**1 |
| M2 | Matrix for coefficients of s**2 |
| M3 | Matrix for coefficients of s**3 |
| M4 | Matrix for coefficients of s**4 |

### Method

Only the non-zero elements of the matrices are printed out.

### Requirements

COMMON blocks: PRNCTL,//
LCAP2 routines: none

## MROOT1

### Identification

SUBROUTINE MROOT1 - Compute Determinant Polynomial Of A Matrix
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Compute polynomial determinant of a matrix with polynomial elements defined by:

$$\underline{M}(s) = \underline{M4}\, s^4 + \underline{M3}\, s^3 + \underline{M2}\, s^2 + \underline{M1}\, s^1 + \underline{M0}$$

This operator is used as one of the steps in evaluating a transfer function via Cramer's method. MROOT1 is called by DETRM and DTERM. When called by DTERM, column substitution with the forcing vector will have been performed already.

### Usage

```
CALL MROOT1(POLY1,ROOT)
```

*POLY1* *output* - Polynomial coefficient array (LCAP2 format)
ROOT output - Polynomial root array (LCAP2 format)

1. Before MROOT1 is used, matrix parameters must first be initialized by calling MINIT0 (once only).

2. Matrix parameters are in COMMON/MATRIX1/. They are to be set before MROOT1 is called. These parameters are defined below:

| Parameter | Preset | Description |
|---|---|---|
| MXM | 1 | Dimension of matrices (1-30) |
| MDEG | 0 | Highest degree of polynomial element (0-4) |
| M0 | 0 | Matrix for coefficients of s**0 |
| M1 | 0 | Matrix for coefficients of s**1 |
| M2 | 0 | Matrix for coefficients of s**2 |
| M3 | 0 | Matrix for coefficients of s**3 |
| M4 | 0 | Matrix for coefficients of s**4 |

### Method

Generalized root finding subroutine MULE and auxiliary subroutine AUXM1 are used to determine the roots of the determinant. As part of the iterative root finding procedure, MULE will supply AUXM1 with a root iterant which AUXM1 will then use to evaluate the matrix

$$\underline{M}(s) = \underline{M4}\, s^4 + \underline{M3}\, s^3 + \underline{M2}\, s^2 + \underline{M1}\, s^1 + \underline{M0}$$

for s = root iterant.

The complex determinant of the above evaluated matrix is then computed by subroutine CXMTX1 and returned to MULE.

The evaluated matrix whose determinant is to be computed by CXMTX1 and the resultant determinant are stored in COMMON/MDET1/. These parameters are described below:

| Parameter | | Description |
|---|---|---|
| NR | input | Dimension of matrix |
| DET | output | Complex determinant |
| NDIMA | input | Row dimension of AMATRIX |
| AMATRIX | input | Complex array of dimension NDIMA x NDIMA |

## Restrictions

The dimension of the matrix must not be greater than 30 x 30. The polynomial elements of the matrix must be degree 4 or less. The degree of the computed polynomial determinant must be less than 50.

MROOT1 is to be used only by Batch LCAP2.

## Requirements

COMMON blocks: CMPOLY,HEADDB,MATRIX1,MDET1,MPAR
LCAP2 routines: AUXM1,AUXP,MULE,PEQUAL,PSYNTH,PZERO,RCLAS,RTPRNO,RZERO

## MROOT2

### Identification

SUBROUTINE MROOT2 - Compute Determinant Polynomial Of A Matrix
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Compute polynomial determinant of a matrix with polynomial elements defined by:

$$\underline{M}(s) = \underline{M4}\, s^4 + \underline{M3}\, s^3 + \underline{M2}\, s^2 + \underline{M1}\, s^1 + \underline{M0}$$

This operator is used as one of the steps in evaluating a transfer function via Cramer's method. MROOT2 is called by IDETRM and IDTERM. When called by IDTERM column substitution with the forcing vector will have been performed already.

### Usage

CALL MROOT2(POLY1,ROOT)

POLY1 output - Polynomial coefficient array (LCAP2 format)
ROOT output - Polynomial root array (LCAP2 format)

1. Before MROOT2 is used, matrix parameters must first be initialized by calling MINTI2 (once only).

2. Matrix parameters are in blank common //. They are to be set before MROOT2 is called. These parameters are defined below:

| Parameter | Preset | Description |
|---|---|---|
| MXM | 1 | Dimension of matrices (1-30) |
| MDEG | 0 | Highest degree of polynomial element (0-4) |
| M0 | 0 | Matrix for coefficients of s**0 |
| M1 | 0 | Matrix for coefficients of s**1 |
| M2 | 0 | Matrix for coefficients of s**2 |
| M3 | 0 | Matrix for coefficients of s**3 |
| M4 | 0 | Matrix for coefficients of s**4 |

### Method

Generalized root finding subroutine MULE and auxiliary subroutine AUXM1 are used to determine the roots of the determinant. As part of the iterative root finding procedure, MULE will supply AUXM1 with a root iterant which AUXM1 will then use to evaluate the matrix

$$\underline{M}(s) = \underline{M4}\, s^4 + \underline{M3}\, s^3 + \underline{M2}\, s^2 + \underline{M1}\, s^1 + \underline{M0}$$

for s = root iterant.

The complex determinant of the above evaluated matrix is then computed by subroutine CXMTX1 and returned to MULE. The evaluated matrix whose determinant is to be computed by CXMTX1 and the resultant determinant is stored in blank common //. These parameters are described below:

| Parameter | | Description |
|---|---|---|
| NR | input | Dimension of matrix |
| DET | output | Complex determinant |
| NDIMA | input | Row dimension of AMATRIX |
| AMATRIX | input | Complex array of dimension NDIMA x NDIMA |

## Restrictions

The dimension of the matrix must not be greater than 30 x 30. The polynomial elements of the matrix must be degree 4 or less. The computed polynomial determinant must be less than 50.

MROOT2 is to be used only by Interactive LCAP2.

## Requirements

COMMON blocks: CMPOLY,HEADDB,INTCOM,MPAR,//
LCAP2 routines: AUXM2,AUXP,MPRINT2,MULE,PEQUAL,PSYNTH,PZERO,RCLAS,RTPRNO, RZERO

# MRXFM

## Identification

SUBROUTINE MRXFM - Multirate Z Plane Transform (In Rational Form) by Sklansky's Frequency Decomposition Method
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Compute multirate z transform, in rational form, by Sklansky's frequency decomposition method.

## Usage

```
CALL MRXFM(XROOTJ,XROOTI,TSLOW,NRATIO)
```

| | | |
|---|---|---|
| XROOTJ | input | - Complex z plane transfer function root array (LCAP2 format) of input at faster sampling rate |
| XROOTI | output | - Complex z plane transfer function root array (LCAP2 format) of output at the slower sampling rate |
| TSLOW | input | - Sampling period of slower output sampler |
| NRATIO | input | - Ratio of slower/input sampling periods |

## Method

Zeroes of the function defined by Sklansky's frequency decomposition method are found by using the generalized root finding subroutine MULE and auxiliary subroutine AXXMRX. To improve the computational accuracy, the calculations are performed in the zeta plane rather than in the z plane. (See description of subroutine ZZETAXM for definition of the zeta plane)

## Restrictions

The degree of the transfer functions must be less than 50.

## Requirements

COMMON blocks: COMAXX,HEADDB,ITEST,TEMPRT
LCAP2 routines: AXXMRN,EJKNI,EVLRRT,EVMRRT,MULE,RCLAS,RREQU,RTPRNO,
XTRACT,ZETAZXM,ZZETAXM

# MULE

## Identification

SUBROUTINE MULE - General Root Finding Subroutine
CDC FORTRAN 4
J. F. Holt
Aerospace Corporation

## Purpose

To determine the zeroes of any analytic function F(Z) using complex arithmetic. Both real and complex roots can be obtained.

## Usage

```
      CALL MULE(KN,N,JG,NP,NJG,NRT,MRT,MXC,MAXIT,JMAX,RXZ,NREV,MAK,
     +IMGZ,RTS,EP1,EP2,EP3,EP4,EP5,ST1,ST2,ST3,SKU,SKL,AUX)
```

```
KN     input  - No. of known or previously computed roots
                These roots must be stored in RTS(1) - RTS(KN)
N      input  - No. of roots to be found for this call to MULE
JG     input  - Initial guess option for the user
NP     input  - output flag. If
                =2, Suppress all internal printing
                =1, Print only the final iteration of each root
                =0, Print all iterants
NJG    input  - Complex conjugate option. If
                =0, Accepts conjugate (if complex) as root
                =1, Do not use conjugate as next guess
NRT    output - No. or roots which actually converged (see Ref. 3 for criteria)
MRT    output - Approx. no. of equal or close roots
MXC    output - No. of roots which iterated MAXIT times
MAXIT  input  - Max. number of iterations allowed per root
JMAX   input  - After JMAX iterations, if conv. .LE. .01 (rel conv.),
                use EP2 (if Z(JMAX) .LE. 1) as new guess. Otherwise,
                use Aitken's delta sq. to compute new guess.
RXZ    input  - Scale factor for special searching procedure after
                JMAX iterations have occurred.
NREV   input  - Option to reverse root(KN+1 thru KN+n) and repeat iteration.
                =1 If (MXC) non-zero, reverse roots and repeat solution
                =0 Exit from subroutine after N roots obtained
MAK    input  - After JMAX iterations perform Aitken delta sq. extrapolation.
                every MAK iterations. Recommend MAK=6 or more.
IMGZ   input  - If .EQ. 0, set imag. part of root to zero
RTS    input  - Root array for guesses if used
       output - Complex array of roots found (dimension of at least KN+N)
EP1    input  - Relative error criterion
EP2    input  - Extremely small guess (abs. value) to be used after JMAX
```

| | | |
|---|---|---|
| | | iterations if current iter. less than 1. |
| EP3 | input | - Criteria for perturbing initial guesses |
| EP4 | input | - Criteria for determining true zero for complex parts |
| EP5 | input | - Criteria for determining absolute zero |
| ST1 | input | - Standard starting value |
| ST2 | input | - Standard starting value |
| ST3 | input | - Standard starting value |
| SKU | input | - Scalar (abs. value) to prevent overflow of F(Z) |
| SKL | input | - Scalar (abs. value) to prevent underflow of F(z) |
| AUX | input | - Function evaluation subroutine (CALL AUX(RT,FRT,SC)) where RT=current iterant, FRT=F(RT), SC=scalar exponent |

1. See listing of MULE for a more complete description of the arguments.

## Method

Uses an improved version of Muller's method which has been combined with AITKEN's delta square extrapolation, an automatic scaling procedure and a simple but efficient searching technique. See Ref. 3 for a more complete description.

## Restrictions

The user must provide an auxiliary subroutine (see AUX) which evaluates the function F(Zi) given iterant Zi. The name of the auxiliary subroutine must be defined as an argument in the calling sequence and also through the use of the EXTERNAL statement in the calling program or subroutine.

## Requirements

COMMON blocks: none
LCAP2 routines: none

# NORM

## Identification

SUBROUTINE NORM - Normalize Transfer Function
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Normalize the coefficients of a transfer function.

## Usage

CALL NORM(PNI,PDI,RTNI,RTDI,INI,IDI)

| | | |
|---|---|---|
| PNI | input | - Polynomial coefficient array (LCAP2 format) of numerator |
| | output | - Normalized polynomial coefficient array (LCAP2 format) of numerator |
| PDI | input | - Polynomial coefficient array (LCAP2 format) of denominator |
| | output | - Normalized polynomial coefficient array (LCAP2 format) of denominator |
| RTNI | input | - Complex polynomial root array (LCAP2 format) of numerator |
| | output | - Normalized complex polynomial root array (LCAP2 format) of numerator |
| RTDI | input | - Complex polynomial root array (LCAP2 format) of denominator |
| | output | - Normalized complex polynomial root array (LCAP2 format) of denominator |
| INI | input | - =0 if numerator coefficients and roots are available<br>=1 if only numerator coefficients are available |
| IDI | input | - =0 if denominator coefficients and roots are available<br>=1 if only denominator coefficients are available |

1. Normalization parameters are in COMMON/HEADDB/. They are to be set before NORM is called. See description for SNORM.

## Restrictions

The degree of the transfer function must be less than 50.

## Requirements

COMMON blocks: INTCOM,HEADDB
LCAP2 routines: none

## OPMESG

### Identification

SUBROUTINE OPMESG - Operation Message
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Print out summary of LCAP2 operations. Message supplied as a Hollerith array in the calling sequence.

### Usage

CALL OPMESG(INTFLG,MESAGE,NLINE)

| | | |
|---|---|---|
| INTFLG | input | - .EQ.0 for batch version, .NE.0 for interactive version |
| MESAGE | input | - Hollerith array of dimension n*5 where n must be equal or greater than the value of NLINE |
| NLINE | input | - Number of lines of message |

1. For the batch version the message will be enclosed by a rectangular box, i.e.,

```
**********************************************
*                                            *
*                                            *
**********************************************
```

2. For the interactive version the message will be preceded by an arrow, i.e.,

```
--->
```

3. If PRNFLG3 (preset=1) of COMMON/PRNCTL/ .EQ.0, printout is suppressed.

### Requirements

COMMON blocks: PRNCTL
LCAP2 routines: none

## OPPRN

### Identification

SUBROUTINE OPPRN - Operand Message
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Print out summary of LCAP2 operations using tables.

### Usage

```
      CALL OPPRN(I,J,K,ICODE,JCODE,KCODE,IOPT,IPLANE,IARG)

I      input  - First argument, i=I
J      input  - Second argument, j=J
K      input  - Third argument, k=K
ICODE  input  - =1   POLY(i) =
                =3   ROOT(i) =
                     (the character - below designates s, w or z
                      as determined by the value of IPLANE)
                =5    -PTF(i)=
                =6  -ROOT(i) =
JCODE  input  - =1              POLY(j)
                =3              ROOT(j)
                =4             ROOTS OF
                =5          PSYNTH(ROOT(j))
                =6              -PTF(j)
                =7             -ROOT(j)
                =8          PSYNTH(-ROOT(J))
                =9           (POLYN/POLYD)
                =10          (ROOTN/ROOTD)
                =11            NUMERATOR OF
                =12          DENOMINATOR OF
                =13             WZXFM OF
                =14             ZWXFM OF
                =15         NORMALIZATION OF
                =16             SWXFM OF
                =17             SZXFM OF
                =18            ZMRXFM OF
                =19            WMRXFM OF
IOPT   input  - =1                          +
                =2                          -
                =3                          *
                =4                          /
                =0                       (blank)
```

```
KCODE  input  - =1                                       POLY(k)
                =3                                       ROOT(k)
                =5                                  PSYNTH(ROOT(k)
                =6                                       -PTF(k)
                =7                                      -ROOT(k)
                =8                                  PSYNTH(-ROOT(k))
IPLANE input  - .EQ.0 for s plane, .LT.0 for w plane, .GT.0 for z plane
IARG   input  - .NE.0 if there is no third part to the printout,
                i.e., no KCODE
```

## Restrictions

If PRNFLG3 (preset=1) of COMMON/PRNCTL/ .EQ.0, printout will be suppressed.

## Requirements

COMMON blocks: INTCOM,PRNCTL
LCAP2 routines: HOLLI

## OSCALE

### Identification

SUBROUTINE OSCALE - Optimum Plot Scale
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Compute optimum plot scale based upon multiple of 2., 2.5, 5. or 10. units per inch.

### Usage

```
CALL OSCALE(YMIN,YMAX,YDELTA,YLNTH)
```

| | | |
|---|---|---|
| YMIN | input | - Minimum value of data to be plotted |
| | output | - Minimum value of optimum scale |
| YMAX | input | - Maximum value of data to be plotted |
| | output | - Maximum value of optimum scale |
| YDELTA | input | - Length of plot scale |
| YLNTH | input | - Full scale in inches |

### Restrictions

YMAX must be .GE. YMIN.

### Requirements

COMMON blocks: none
LCAP2 routines: none

# PACK

## Identification

SUBROUTINE PACK - Pack Root Information Into Word
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Pack root information into the real part of a complex variable. The first word of a complex root array (LCAP2 format) contains this packed information.

## Usage

```
      CALL PACK(ROOT,NUM,ICOM,IR,IZ)

ROOT  input  - Complex variable
      output - Complex variable with real part containing packed
               root information (imaginary part unchanged)
NUM   input  - Total Number of roots
ICOM  input  - Number of complex roots
IR    input  - Number of real roots not at the origin
IZ    input  - Number of roots at the origin
```

## Method

The first word of a complex array, in LCAP2 format, is used to store information characterizing a polynomial. The real part of this word is packed as: bit 1 is the unit digit, bit 2 is the tens digit, .. etc.

| Decimal Digit | Description |
|---|---|
| 1,2 | Total number of roots |
| 3,4 | Number of complex roots (an even number) |
| 5,6 | Number of real roots not at the origin |
| 7,8 | Number of roots at the origin |

```
Example:  NUM = 8
          ICOM= 4
          IR  = 3
          IZ  = 1
          CALL PACK(ROOT,NUM,ICOM,IR,IZ)
```

would yield, value of REAL(ROOT(1)) = 103048

The imaginary part of the first word of a complex root array in LCAP2 format is the low order non-zero coefficient of the polynomial. This value is not affected by this subroutine.

The code for this routine is in subroutine XTRACT.

Requirements

COMMON blocks: none
LCAP2 routines: none

# PADD

## Identification

SUBROUTINE PADD - LCAP2 Operator, Polynomial Add
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Add two polynomials using LCAP2 indices.

## Usage

```
CALL PADD(I,J,K)
```

I input - Index of resultant polynomial sum
J input - Index of first polynomial to be added
K input - Index of second polynomial to be added

## Restrictions

The degree of the polynomials must be less than 50.

## Requirements

COMMON blocks: PRNCTL,TFTEMP
LCAP2 routines: ADDP,ENDLINE,FETPY,OPPRN,PROOT,PYPRN1,PYPRN4,STRPY

# PCHEK

## Identification

SUBROUTINE PCHEK - Polynomial Check For Extraneous Coefficients
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Check polynomial coefficient array to see if the degree is between 0 and 49. If it is not, zero out all elements of the array.

Also zero out undefined coefficients of the array which correspond to coefficients larger than the degree of the polynomial.

## Usage

```
CALL PCHEK(P)
```

```
P  input  - Polynomial coefficient array (LCAP2 format)
   output - Polynomial coefficient array (LCAP2 format) with undefined
            coefficients set to zero.
```

## Requirements

COMMON blocks: none
LCAP2 routines: PZERO

# PEQU

## Identification

SUBROUTINE PEQU - LCAP2 Operator, Polynomial Equal
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Equate polynomials using LCAP2 indices.

## Usage

```
CALL PEQU(I,J)
```

I input - Index of resultant polynomial
J input - Index of polynomial to be equated with

## Restrictions

The degree of the polynomials must be less than 50.

## Requirements

COMMON blocks: TFTEMP
LCAP2 routines: ENDLINE,FETPY,OPPRN,PEQUAL,PYPRN1,PYPRN4,RTEQU,STRPY

# PEQUAL

## Identification

SUBROUTINE PEQUAL - Equate Polynomial Coefficient Arrays
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Equate polynomials in coefficient form.

## Usage

```
CALL PEQUAL(P1,P2)
```

P1 input - Polynomial coefficient array (LCAP2 format)
P2 output - Polynomial coefficient array (LCAP2 format)

## Restrictions

The degree of the polynomials must be less than 50.

## Requirements

COMMON blocks: none
LCAP2 routines: none

## PLDC

### Identification

SUBROUTINE PLDC - LCAP2 Operator, Polynomial Load In Coefficient Form
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Load coefficients into polynomial coefficient array, POLYi.

### Usage

CALL PLDC(I)

I input - Index where polynomial is to be stored

1. Polynomial coefficients are entered with polynomial array POLY (LCAP2 format) which are in COMMON/HEADDB/. They are to be set before PLDC is called.

2. The calling program must include COMMON/HEADDB/ and the appropriate DIMENSION and EQUIVALENCE statements for POLY.

3. The roots of POLYi will not be automatically computed. If this is desired, follow this operation with the operator PRTS(I).

### Restrictions

The degree of the polynomials must be less than 50.

### Requirements

COMMON blocks: INTCOM,HEADDB,TFTEMP
LCAP2 routines: ENDLINE,OPMESG,PEQUAL,PPRN1,PYPRN4,STRPY

# PLDR

## Identification

SUBROUTINE PLDR - LCAP2 Operator, Polynomial Load In Root Form
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Load roots in polynomial root array, ROOT. After the roots have been loaded, the coefficients of the polynomial are computed and stored in the polynomial coefficient array POLYi.

## Usage

```
CALL PLDR(I)
```

I input - Index where polynomial is to be stored

1. Polynomial roots are entered with polynomial root array ROOT (LCAP2 format) which is in COMMON/HEADDB/. They are to be set before PLDR is called.

2. The calling program must include COMMON/HEADDB/ and the appropriate DIMENSION and EQUIVALENCE statements for ROOT.

## Restrictions

The degree of the polynomials must be less than 50.

## Requirements

COMMON blocks: INTCOM,HEADDB,TFTEMP
LCAP2 routines: ENDLINE,OPMESG,PSYNTH,PYPRN4,RTEQU,RTPRN2,STRPY

## PLQ

### Identification

SUBROUTINE PLQ - Synthesize Polynomial From Lags and Quadratics
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Synthesize or compute polynomial from a product of first order lags and quadratics.

### Usage

CALL PLQ(NTAU,TAU,NQUAD,OMEGA,ZETA,POLY)

| | | |
|---|---|---|
| NTAU | input | - Number of first order lags |
| TAU | input | - Array of time constants (dimension must be at least the value of NTAU) |
| NQUAD | input | - Number of quadratic terms |
| OMEGA | input | - Array of quadratic frequencies (dimension must be at least the value of NQUAD) |
| ZETA | input | - Array of quadratic damping coefficients (dimension at least the value of NQUAD) |
| POLY | output | - Polynomial coefficient array (LCAP2 format) |

### Restrictions

The degree of the polynomial formed must be less than 50.

### Requirements

COMMON blocks: none
LCAP2 routines: PMULT,PZERO

# PMULT

## Identification

SUBROUTINE PMULT - Polynomial Multiply (Coefficient Form)
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Multiply two polynomials in coefficient form.

## Usage

```
CALL PMULT(A,B,C)
```

A input - Polynomial coefficient array (LCAP2 format)
B input - Polynomial coefficient array (LCAP2 format)
C output - Polynomial coefficient array (LCAP2 format) of product

## Method

Double precision is used for the intermediate calculations.

## Restrictions

The degree of the polynomials must be less than 50.

## Requirements

COMMON blocks: none
LCAP2 routines: none

# PMPY

## Identification

SUBROUTINE PMPY - LCAP2 Operator, Polynomial Multiply
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Multiply two polynomials using LCAP2 indices.

## Usage

```
CALL PMPY(I,J,K)
```

I input - Index of resultant polynomial product
J input - Index of polynomial multiplicand
K input - Index of polynomial multiplier

## Method

If only the coefficients of the j-th and k-th polynomials are available, the product is computed by multiplication of the coefficients. If the roots of the j-th and k-th polynomials are available, the product is computed by combining the roots. The coefficients of the product are then formed from these roots.

## Restrictions

The degree of the polynomials must be less than 50.

## Requirements

COMMON blocks: TFTEMP
LCAP2 routines: ENDLINE,FETPY,OPPRN,PMULT,PSYNTH,PYPRN1,PYPRN4,RTMPY,
STRPY

## PPADD

### Identification

SUBROUTINE PPADD - Transfer Function Addition (Coefficient Form)
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Add two transfer functions, in coefficient form, by rationalization.

### Usage

```
CALL PPADD(R1,R2,R3)
```

R1 input - Transfer function coefficient array (LCAP2 format)
R2 input - Transfer function coefficient array (LCAP2 format)
R3 output - Transfer function coefficient array (LCAP2 format) of sum

### Method

After rationalization is completed, a check is made to determine if there are any common roots at the origin between the numerator and the denominator. If there are any, they are cancelled.

### Restrictions

The degree of the transfer functions must be less than 50.

### Requirements

COMMON blocks: HEADDB
LCAP2 routines: ADDP,PEQUAL,PMULT,SUBP

# PPEQU

## Identification

SUBROUTINE PPEQU - Equate Transfer Function Arrays
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Equate transfer functions in coefficient form.

## Usage

CALL PPEQU(P1,P2)

P1 input - Transfer function array (LCAP2 format)
P2 output - Transfer function array (LCAP2 format)

## Restrictions

The degree of the polynomials must be less than 50.

The code for this routine is in subroutine PEQUAL.

## Requirements

COMMON blocks: none
LCAP2 routines: none

# PPMPY

## Identification

SUBROUTINE PPMPY - Multiply Two Transfer Functions (Coefficient Form)
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Multiply two transfer functions in coefficient form.

## Usage

```
CALL PPMPY(A,B,C)
```

A input - Transfer function coefficient array (LCAP2 format)
B input - Transfer function coefficient array (LCAP2 format)
C output - Transfer function coefficient array (LCAP2 format) of product

## Method

Two separate calls to subroutine PMULT are used to compute the numerator and denominator.

## Restrictions

The degree of the transfer functions must be less than 50.

## Requirements

COMMON blocks: none
LCAP2 routines: PMULT

# PPRINT

## Identification

SUBROUTINE PPRINT - Print Out Polynomial Coefficients With Integer Identifier
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Print out polynomial coefficients with integer identifier.

## Usage

```
CALL PPRINT(POUT,IDENT)
```

POUT   input  - Polynomial coefficient array (LCAP2 format)
IDENT  input  - Identifier (.GT.0) used for labeling printout.

## Method

Coefficients are printed out in ascending order.

## Restrictions

The degree of the polynomial must be less than 50.

## Requirements

COMMON blocks: none
LCAP2 routines: none

# PPRN

## Identification

SUBROUTINE PPRN - LCAP2 Operator, Print Out Polynomial
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Print out polynomial using an LCAP2 index.

## Usage

```
CALL PPRN(I)
```

I input - Index of polynomial to be printed out

## Method

Roots of the polynomial are printed out only if they defined ( previously computed or loaded in). Coefficients of the polynomial are printed out in ascending order.

## Restrictions

The degree of the polynomial must be less than 50.

## Requirements

COMMON blocks: INTCOM,TFTEMP
LCAP2 routines: ENDLINE,FETPY,OPMESG,PYPRN4

## PPRN1

### Identification

SUBROUTINE PPRN1 - Print Out Polynomial Coefficient Array
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Print out polynomial coefficient array with identifiers using tables.

### Usage

```
CALL PPRN1(IFLAG,POLY,IDENT,IPLANE)
```

```
IFLAG   input  - =0 No heading
                 =1 DEGREE OF POLY(I) IS
                 =2 DEGREE OF POLY is
                 =3 DEGREE OF NUMERATOR OF -PTF(I) IS
                    DEGREE OF DENOMINATOR OF -PTF(I) IS
                    (the character - above is s, w or z as
                    determined by the value of IPLANE)
POLY    input  - Polynomial coefficient array (LCAP2 format)
IDENT   input  - Identifier (0-99) used for labeling printout
IPLANE  input  - =0 for s plane, =-1 for w plane, =1 for z plane
```

### Requirements

COMMON blocks: none
LCAP2 routines: HOLLI

# PPSUB

## Identification

SUBROUTINE PPSUB - Transfer Function Subtraction (Coefficient Form)
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Subtract two transfer functions, in coefficient form, by rationalization.

## Usage

```
CALL PPSUB(R1,R2,R3)
```

| | | |
|---|---|---|
| R1 | input | - Transfer function coefficient array (LCAP2 format) (minuend) |
| R2 | input | - Transfer function coefficient array (LCAP2 format) (subtrahend) |
| R3 | output | - Transfer function coefficient array (LCAP2 format) of difference |

## Method

After rationalization is completed, a check is made to determine if there are any common roots at the origin between the numerator and the denominator. If there are any, they are cancelled.

The code for this routine is in subroutine PPADD.

## Restrictions

The degree of the transfer functions must be less than 50.

## Requirements

COMMON blocks: HEADDB
LCAP2 routines: ADDP,PEQUAL,PMULT,SUBP

# PPZERO

## Identification

SUBROUTINE PPZERO - Zero Out Transfer Function Coefficient Array
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Zero out polynomial coefficient array.

## Usage

```
CALL PPZERO(P)
```

P input - Transfer function coefficient array (LCAP2 format)
output - Transfer function coefficient array (LCAP2 format) zeroed out

## Method

The code for this routine is in subroutine PZERO.

## Requirements

COMMON blocks: none
LCAP2 routines: none

# PROOT

## Identification

SUBROUTINE PROOT - Polynomial Root Finder
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Find roots of a polynomial.

## Usage

```
CALL PROOT(POLY,ROOT)
```

POLY input - Polynomial coefficient array (LCAP2 format)
ROOT output - Complex polynomial root array (LCAP2 format)

## Method

Generalized root finding subroutine MULE and auxiliary subroutine AUXP are used to determine the roots. Before MULE is called to compute the roots of the polynomial, array POLY must first be copied into POLYC of COMMON/CMPOLY/ so that MULE and AUXP has the proper interface.

## Restrictions

The degree of the polynomial must be less than 50.

## Requirements

COMMON blocks: CMPOLY,HEADDB
LCAP2 routines: AUXP,MULE,PEQUAL,PPRN1,RCLAS,RTPRNO,RZERO

# PRTS

## Identification

SUBROUTINE PRTS - LCAP2 Operator, Find Roots Of A Polynomial
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Find roots of a polynomial using an LCAP2 index.

## Usage

```
CALL PRTS(i)
```

I  input  - Index of polynomial

## Restrictions

If the roots of POLYi were previously computed or loaded in, the program will not recompute the roots from the coefficients. A message to this effect will be printed.

The degree of the polynomial must be less than 50.

## Requirements

COMMON blocks: INTCOM,TFTEMP
LCAP2 routines: ENDLINE,FETPY,HOLLI,OPMESG,PROOT,PYPRN1,PYPRN4,STRPY

# PSUB

## Identification

SUBROUTINE PSUB - LCAP2 Operator, Polynomial Subtract
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Subtract two polynomials using LCAP2 indices.

## Usage

```
CALL PSUB(I,J,K)
```

I input - Index of resultant polynomial difference
J input - Index of first polynomial (minuend)
K input - Index of second polynomial (subtrahend)

## Restrictions

The degree of the polynomial must be less than 50.

## Requirements

COMMON blocks: TFTEMP
LCAP2 routines: ENDLINE,FETPY,OPPRN,PROOT,PYPRN1,PYPRN4,STRPY

# PSYNTH

## Identification

SUBROUTINE PSYNTH - Synthesize Polynomial Coefficients From The Roots
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Synthesize or compute the coefficients of a polynomial from its roots.

## Usage

```
CALL PSYNTH(ROOT,POLY)
```

ROOT input - Complex polynomial root array (LCAP2 format)
POLY output - Polynomial coefficient array (LCAP2 format)

## Method

Double precision is used for intermediate calculations.

## Restrictions

The degree of the polynomial must be less than 50.

## Requirements

COMMON blocks: none
LCAP2 routines: PACK,RCLAS,XTRACT

## Alternate Entry

SUBROUTINE PSYNTH1 - Same as PSYNTH except that is assumes that the root array has already been classified by a prior call to subroutine RCLAS.

# PYPRN1

## Identification

SUBROUTINE PYPRN1 - Print Out Polynomial Coefficients and Roots
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Print out polynomial coefficients and roots (if available) under control of PRNFLG1.

## Usage

```
CALL PYPRN1(I,POLY,ROOT,IN)
```

```
I     input  - Identifier (0-99) used for labeling printout
POLY  input  - Polynomial coefficient array (LCAP2 format)
ROOT  input  - Complex polynomial root array (LCAP2 format)
IN    input  - =1 for coefficient form only
               =0 for coefficient and root form
```

1. If PRNFLG1 (preset=1) in COMMON/PRNCTL/ .EQ.0, printout is suppressed.

## Method

Uses subroutine RTPRN2 to print out the roots and subroutine PPRN1 to print out the coefficients.

## Restrictions

The degree of the polynomial must be less than 50.

## Requirements

COMMON blocks: PRNCTL
LCAP2 routines: none

# PYPRN4

## Identification

SUBROUTINE PYPRN4 - *Print Out Polynomial Coefficients and Roots*
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Print out polynomial coefficients and roots (if available) under control of PRNFLG4.

## Usage

```
      CALL PYPRN4(I,POLY,ROOT,IN)

I     input  - Identifier (0-99) used for labeling printout
POLY  input  - Polynomial coefficient array (LCAP2 format)
ROOT  input  - Complex polynomial root array (LCAP2 format)
IN    input  - =1 for coefficient form only
               =0 for coefficient and root form
```

1. If PRNFLG4 (preset=1) in COMMON/PRNCTL/ .EQ.0, printout is suppressed.

## Method

Uses subroutine RTPRN2 to print out the roots and subroutine PPRN1 to print out the coefficients.

## Restrictions

The degree of the polynomial must be less than 50.

## Requirements

COMMON blocks: PRNCTL
LCAP2 routines: none

# PZERO

## Identification

SUBROUTINE PZERO - Zero Out Polynomial Coefficient Array
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Zero out polynomial coefficient array.

## Usage

```
CALL PZERO(P)
```

P input - Polynomial coefficient array (LCAP2 format)
output - Polynomial coefficient array (LCAP2 format) zeroed out

## Requirements

COMMON blocks: none
LCAP2 routines: none

# RCLAS

## Identification

SUBROUTINE RCLAS - Classify Polynomial Root Array
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Classify polynomial roots by complex roots, non-zero real roots and zero roots.

## Usage

```
CALL RCLAS(ROOTS,ICOM,IR,IZ)
```

```
ROOTS  input  - Complex polynomial root array (LCAP2 format)
       output - Complex polynomial root array (LCAP2 format) with
                roots classified
ICOM   output - Number of complex roots found (an even number)
IR     output - Number of non-zero real roots found
IZ     output - Number of zero roots (at the origin)
```

1. The parameters ERCNJ, ERCX and ERCZ (preset=1.E-4, 1.E4 and 1.E5, respectively) used in the classification of the roots are in COMMON/HEADDB/.

2. The real part of ROOTS(1) is a packed word containing (1) total number of roots, (2) number of complex roots, (3) number of non-zero real roots and (4) number of zero roots. See description of XTRACT for details.

3. If PRN3 (preset=0) of COMMON/HEADDB/ .NE.0, diagnostic printout will be produced.

## Method

1. A root is considered complex if ABS(real/imag).LT.ERCX for imag.NE.0.

2. If imag.NE.0 and ABS(real/imag).GT.ERCX, imag part will be set to zero.

3. If real.NE.0 and ABS(imag/real).GT.ERCX, real part will be set to zero.

4. Each pair of complex roots is checked to see if the roots are conjugates. They are conjugated if the following are true:

```
SQRT((REAL(ROOTS(I))-REAL(ROOTS(I+1)))**2 + (AIMAG(ROOTS(I))
+ AIMAG(ROOTS(I+1)))**2).LE.ERCNJ
```

5. A root is considered to be zero if its absolute value is .LT.ERCZ.

Restrictions

The degree of the polynomial must be less than 50.

Requirements

COMMON blocks: HEADDB
LCAP2 routines: PACK,RTPRNO

## REMARKI,(I=1,5)

### Identification

SUBROUTINE REMARKi - Print Out Remarks, (i=1,5)
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Single FORTRAN statement for printing out Hollerith data.

### Usage

CALL REMARKi(A)

```
A  input  - Hollerith data with format 10H... if i=1
                                       20H... if i=2
                                       30H... if i=3
                                       40H... if i=4
                                       50H... if i=5
```

1. Example: CALL REMARK2(20HTHIS IS AN EXAMPLE  ) will print out,

   THIS IS AN EXAMPLE

### Requirements

COMMON blocks: none
LCAP2 routines: none

# RESDU

## Identification

SUBROUTINE RESDU - Residues For Partial Fraction Expansion
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Compute residues for partial fraction expansion.

## Usage

```
CALL RESDU(MUI,IPCOM,IPR,IPZ,QROT,PROT,GAMMA,LAMBDA,XKRL)
```

| | | |
|---|---|---|
| MUI | input | - Number of poles at the origin |
| IPCOM | input | - Number of complex poles (an even number) |
| IPR | input | - Number of real (non-zero) poles |
| IPZ | input | - Number of poles at the origin |
| QROT | input | - Complex polynomial root array (LCAP2 format) for num. |
| PROT | input | - Complex polynomial root array (LCAP2 format) for denom. |
| GAMMA | output | - Complex array of residues for the poles not at the origin |
| LAMBDA | output | - Complex array of residues for the poles at the origin |
| XKRL | input | - Root locus gain |

## Restrictions

The degree of the transfer function must be less than 50.

The number of non-zero poles must not be greater than the order of the numerator.

No multiple poles are allowed except for those at the origin. Multiple poles not at the origin can be represented by distinct poles displaced from each other by a small amount.

The number of multiple poles at the origin must be 5 or less.

## Requirements

COMMON blocks: HEADDB
LCAP2 routines: none

# RESTORE

## Identification

SUBROUTINE RESTORE - Restore Polynomial, Transfer Function And Matrix Data
CDC FORTRAN 4
C. L. Wong and E. A. Lee
Aerospace Corporation

## Purpose

Restore polynomial, transfer function and matrix data from a previous batch or interactive job for a restart capability in Batch LCAP2.

## Usage

CALL RESTORE(IPRNFG)

IPRNFG input - =0 for no printout of restored data

## Method

Reads in data stored on TAPE30 and copies it into COMMON/SCMBLK/ and the sequential files TAPE84, TAPE85, TAPE86, TAPE87 and TAPE83. For more details see description for STORE.

## Restrictions

File type for TAPE30 must be declared with 'FILE,TAPE30,BT=I.' File TAPE30 must be attached before executing LCAP2.

## Requirements

COMMON blocks: ACOM,HEADDB,PRNCTL,MATRIX1,MDET1,SCMBLK,TFPCNT,//
LCAP2 routines: BPRINT1,ENDLINE,MPRINT1,OPMESG,PYPRN1,TFPRN1

# RLOCIN1

## Identification

SUBROUTINE RLOCIN1 - Interactive Input Routine For Root Locus
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Interactive input routine for root locus.

## Usage

```
CALL RLOCIN1(IUNIT,IEXIT,NIT,XL,XR,YTOP,YBOT,IPRN1)
```

| | | |
|---|---|---|
| IUNIT | input | - Tape unit used to save sets of data computed |
| IEXIT | output | - =1 on exit from RLOCIN2 if user wants to reenter root locus parameters<br>=2 on exit from RLOCIN2 if user does not want a plot |
| NIT | output | - Number of records written on TAPE IUNIT |
| XL | input | - Auto scaled min x |
| XR | input | - Auto scaled max x |
| YTOP | input | - Auto scaled max y |
| YBOT | input | - Auto scaled min y |
| IPRN1 | output | - =1 on exit from RLOCIN1 if user wants to suppress tabular printout of the roots, =0 otherwise |

## Method

Subroutines RLOCIN1 and RLOCIN2 are called by RLOCUS1. RLOCIN1 is used for inputing root locus gain parameters and for selecting method for incrementing the gains. Subroutine RLOCIN2 is used for inputing plot parameters.

## Requirements

COMMON blocks: HEADDB,//
LCAP2 routines: GRAF1,ITITLE

## Alternate Entry

SUBROUTINE RLOCIN2 - Interactive input routine for root locus.

## RLOCUS1

### Identification

SUBROUTINE RLOCUS1 - Root Locus
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Compute and plot root locus. This routine is called by SLOCI, WLOCI and ZLOCI.

### Usage

CALL RLOCUS1(IPLANE,INDXI,PN,PD,RTN,RTD,INI,IDI,XPOINT,YPOINT)

| | | |
|---|---|---|
| IPLANE | input | - =0 for s plane, =-1 for w plane, =1 for z plane |
| INDXI | input | - not used anymore |
| PN | input | - Polynomial coefficient array (LCAP2 format) of numerator |
| PD | input | - Polynomial coefficient array (LCAP2 format) of denominator |
| RTN | input | - Complex polynomial root array (LCAP2 format) of numerator |
| RTD | input | - Complex polynomial root array (LCAP2 format) of denominator |
| INI | input | - =0 if numerator coefficients and roots are available<br>=1 if only numerator coefficients are available |
| IDI | input | - =0 if denominator coefficients and roots are available<br>=1 if only denominator coefficients are available |
| XPOINT | output | - Array of x component of root locus points (DIMENSION=1500) |
| YPOINT | output | - Array of y component of root locus points (DIMENSION=1500) |

1. For the batch version, the root locus parameters are in COMMON/HEADDB/. These parameters are to be set before this subroutine is called.

2. For the interactive version, the user is prompted for the root locus parameters.

3. Flag INTFLG (preset=0) of COMMON/INTCOM/ is used to determine if program is in interactive or batch mode. If .NE.0, program is in the interactive mode.

### Method

The root locus is computed by evaluating the roots of the polynomial (PN + GAIN*PD) where GAIN is the varied gain and PN and PD are the numerator and denominator polynomials, respectively.

File TAPE19 is used to save root locus plot points.

If in the interactive mode, RLOCIN1 and RLOCIN2 are called.

Restrictions

The degree of the transfer function must be less than 50.

Requirements

COMMON blocks: AWORDS,HEADDB,INTCOM,PLOT1
LCAP2 routines: ADDP,ANOTAT,DATE,DAYPRN,ELPLOT1,GRAF1,OPPRN,PROOT,
PZERO,RLOCIN1,RLOCIN2,RTEQU,RTPRNO

# RPRINT

## Identification

SUBROUTINE RPRINT - Print Out Transfer Function Coefficient With Integer Identifier
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Print out transfer function coefficients with an integer identifier. (This is a very old routine which will eventually be phased out)

## Usage

```
CALL RPRINT(R,I)
```

R input - Transfer function coefficient array (LCAP2 format)
I input - Identifier (.GT.0) used for labeling printout

## Restrictions

The degree of the transfer function must be less than 50.

## Requirements

COMMON blocks: none
LCAP2 routines: none

## RRTADD

### Identification

SUBROUTINE RRTADD - Add Transfer Functions (Root Form)
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Add two transfer functions in root form.

### Usage

```
CALL RRTADD(RTJ,RTK,RTI,PI)
```

| | | |
|---|---|---|
| RTJ | input | - Complex transfer function root array (LCAP2 format) |
| RJK | input | - Complex transfer function root array (LCAP2 format) |
| RTI | output | - Complex transfer function root array (LCAP2 format) of sum |
| PI | output | - Transfer function coefficient root array (LCAP2 format) of sum |

1. Flag PRN8 (preset=0) of COMMON/HEADDB/ if .NE.0, will print out common roots eliminated.

### Method

Common roots, if any, are first factored from the denominator of the j-th and k-th transfer function. The transfer functions are then added and rationalized. The factored common denominator roots, if any, are then recombined. Subroutine CRELIM is called to remove any common roots between the numerator and denominator. Finally, the coefficients of the resultant transfer function are computed from the roots.

### Requirements

COMMON blocks: HEADDB
LCAP2 routines: ADDP,CRELIM,PROOT,PSYNTH,RPRN,RTEQU,RTMPY,SUBP

### Alternate Entry

SUBROUTINE RRTAD1 - Same as RRTADD except that it will bypass elimination of common roots of the final sum. (RRTAD1 is called by WTRANS)

# RRTEQU

## Identification

SUBROUTINE RRTEQU - Equate Transfer Function Root Arrays
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Equate transfer functions in root form.

## Usage

```
CALL RRTEQU(R1,R2)
```

R1 input - Complex transfer function root array (LCAP2 format)
R2 output - Complex transfer function root array (LCAP2 format)

## Method

The code for this routine is in subroutine RTEQU.

## Restrictions

The degree of the transfer function must be less than 50.

## Requirements

COMMON blocks: none
LCAP2 routines: none

## RRTSUB

### Identification

SUBROUTINE RRTSUB - Subtract Transfer Functions (Root Form)
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Subtract two transfer functions in root form.

### Usage

```
CALL RRTSUB(RTJ,RTK,RTI,PI)
```

| | | |
|---|---|---|
| RTJ | input | - Complex transfer function root array (LCAP2 format) (minuend) |
| RJK | input | - Complex transfer function root array (LCAP2 format) (subtrahend) |
| RTI | output | - Complex transfer function root array (LCAP2 format) of difference |
| PI | output | - Transfer function coefficient root array (LCAP2 format) of difference |

1. Flag PRN8 (preset=0) of COMMON/HEADDB/ if .NE.0, will print out common roots eliminated.

### Method

Common roots, if any, are first factored from the denominator of the j-th and k-th transfer function. The transfer functions are then subtracted and rationalized. The factored common denominator roots, if any, are then recombined. Subroutine CRELIM is called to remove any common roots between the numerator and denominator. Finally, the coefficients of the resultant transfer function are computed from the roots.

The code for this routine is in subroutine RRTADD.

### Requirements

COMMON blocks: HEADDB
LCAP2 routines: ADDP,CRELIM,PROOT,PSYNTH,RPRN,RTEQU,RTMPY,SUBP

# RRTPRN

## Identification

SUBROUTINE RRTPRN - Print Out Transfer Function Roots With Hollerith Identifier
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Print out transfer function root array with a Hollerith Identifier.

## Usage

```
CALL RRTPRN(TFR,WORD)
```

TFR   input  - Complex transfer function root array (LCAP2 format)
WORD  input  - Hollerith word used to label the transfer function

## Restrictions

The degree of the transfer function must be less than 50.

## Requirements

COMMON blocks: none
LCAP2 routines: XTRACT

## Alternate Entries

SUBROUTINE TFRPRN - Same as RRTPRN.

# RRZERO

## Identification

SUBROUTINE RRZERO - Zero Out Transfer Function Root Array
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Initialize transfer function root array to zero.

## Usage

```
CALL RRZERO(R)
```

R  input  - Complex transfer function root array (LCAP2 format)
   output - Complex transfer function root array (LCAP2 format)

## Method

The code for this routine in is subroutine RZERO.

## Requirements

COMMON blocks: none
LCAP2 routines: none

# RTCMNT

## Identification

SUBROUTINE RTCMNT - Root Comment For Interactive Use
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Print comment to interactive user on format used to enter root data. If user desires, program will print out an example.

## Usage

CALL RTCMNT

## Requirements

COMMON blocks: ACOM
LCAP2 routines: none

## RTEQU

### Identification

SUBROUTINE RTEQU - Equate Polynomial Root Array
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Equate polynomials in root form.

### Usage

```
CALL RTEQU(R1,R2)
```

R1  input  - Complex polynomial root array (LCAP2 format)
R2  output - Complex polynomial root array (LCAP2 format)

### Restrictions

The degree of the polynomial must be less than 50.

### Requirements

COMMON blocks: none
LCAP2 routines: none

### Alternate Entry

SUBROUTINE REQU - Same as RTEQU

## RTEQ1

### Identification

SUBROUTINE RTEQ1 - Find Number Of Real Roots Equal To (1.,0.)
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Find number of real roots which are equal or very close to (1.,0.). These roots are moved to the end of the list of non-zero roots. This routine is used by subroutine ZWTRANS.

### Usage

CALL RTEQ1(ROOT,L)

ROOT input - Complex polynomial root array (LCAP2 format)
L output - Number of roots found equal to (1.,0.)

### Requirements

COMMON blocks: HEADDB
LCAP2 routines: XTRACT

### Alternate Entry

SUBROUTINE RTEQ2 - Same as RTEQ1 except that the criteria for roots equal to or very close to (-1.,0.) instead of (1.,0.).

# RTMPY

## Identification

SUBROUTINE RTMPY - Multiply Polynomials (Root Form)
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Multiply two polynomials in root form.

## Usage

```
CALL RTMPY(ROOT1,ROOT2,ROOT3)
```

ROOT1 input - Complex polynomial root array (LCAP2 format)
ROOT2 input - Complex polynomial root array (LCAP2 format)
ROOT3 output - Complex polynomial root array (LCAP2 format) of product

## Method

The product ROOT3 is obtained by collecting the roots from ROOT1 ROOT2.

## Restrictions

The degree of the polynomials must be less than 50. If the degree of the resultant polynomial is greater than 49, the program will be terminated.

## Requirements

COMMON blocks: none
LCAP2 routines: RCLAS,REQU

## Alternate Entry Name

SUBROUTINE RTADD - Same as RTMPY.

# RTPRN

## Identification

SUBROUTINE RTPRN - Print Out Polynomial Roots With Hollerith Identifier
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Print out polynomial root array with a Hollerith Identifier.

## Usage

```
CALL RTPRN(TFR,WORD)
```

TFR input - Complex polynomial root array (LCAP2 format)
WORD input - Hollerith word used to label the polynomial

## Method

The code for this routine is in subroutine RRTPRN.

## Restrictions

The degree of the polynomial must be less than 50.

## Requirements

COMMON blocks: none
LCAP2 routines: EXTRACT

# RTPRNO

## Identification

SUBROUTINE RTPRNO - Print Out Roots Of A Polynomial
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Print out roots of a polynomial.

## Usage

CALL RTPRNO(ROOT)

ROOT input - Complex polynomial root array (LCAP2 format)

## Method

The printout has the following form:

| NO | REAL | IMAG | OMEGA | ZETA |
|---|---|---|---|---|
| 1 | ... | ... | ... | ... |
| 2 | ... | ... | ... | ... |
| . | . | . | . | . |
| . | . | . | . | . |

## Restrictions

The roots in the array ROOT are assumed to have been classified by a prior call to RCLAS. If not, the complex roots may not have their omega and zeta values printed out.

## Requirements

COMMON blocks: none
LCAP2 routines: XTRACT

# RTPRN2

## Identification

SUBROUTINE RTPRN2 - Print Out Polynomial Root Array
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Print out polynomial or transfer function roots with identifiers using tables.

## Usage

```
     CALL RTPRN2(IFLAG,ROOT,IDENT,IPLANE)

IFLAG   input  - =0 no heading
                 =1 THE ROOTS OF ROOT(IDENT) ARE
                 =2 THE ROOTS OF ROOT ARE
                 =3 THE NUMERATOR ROOTS OF -ROOT(IDENT) ARE
                    THE DENOMINATOR ROOTS OF -ROOT(IDENT) ARE
                 =4 THE ROOTS OF POLY(IDENT) ARE
                    (The character - above designates s, w or z
                    as determined by the value of IPLANE)
                 =5 THE ROOTS OF POLY ARE
ROOT    input  - Complex polynomial root array (LCAP2 format)
IDENT   input  - root array identifier (1-99)
IPLANE  input  - =0 for s plane, =-1 for w plane, =1 for z plane
```

## Requirements

COMMON blocks: none
LCAP2 routines: FCNW2,HOLLI,RTPRN0

# RZERO

## Identification

SUBROUTINE RZERO - Zero Out Polynomial Root Array
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Initialize polynomial root array elements to zero.

## Usage

```
CALL RZERO(R)
```

```
R  input  - Complex polynomial root array (LCAP2 format)
   output - Complex polynomial root array (LCAP2 format)
```

## Requirements

COMMON blocks: none
LCAP2 routines: none

# SELCR

## Identification

SUBROUTINE SELCR - LCAP2 Operator, Eliminate Common Roots Of S Plane
Transfer Function
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Eliminate common roots from an s plane transfer function using an LCAP2 index.

## Usage

```
CALL SELCR(I)
```

I input - Index of s plane transfer function

1. Common root elimination parameters ECRE1 (preset=2.E-4) and ECRE2 (preset=1.E-8) are in COMMON/HEADDB/.

## Method

If a numerator root nrt and a denominator root drt are found such that ABS(drt/nrt - (1.,0.)).LT.ECRE1 for nrt.NE.0 or ABS(drt).LT.ECRE2 for nrt.EQ.0, roots nrt and drt are considered to be common roots and will be eliminated from the transfer function.

## Requirements

COMMON blocks: INTCOM,PRNCTL,TFTEMP
LCAP2 routines: CRELIM,ENDLINE,FCNPLN,FETTFX,OPMESG,OPPRN,PROOT,PSYNTH,
RREQU,RTPRN0,STRTFX,TFPRN1,TFPRN4

## SFAUX

### Identification

COMPLEX FUNCTION SFAUX - Evaluate S Plane Transfer Function Coefficient Array
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Evaluate s plane transfer function coefficient array (LCAP2 format) for use in computing the frequency response. This complex function can be be used by subroutine FREQS1 or FREQS2 to evaluate the transfer function specified by its first argument. It can also be used by user-supplied subroutines similar to SFAUX1.

### Usage

SFAUX(TFC)

TFC input - Transfer function coefficient array (LCAP2 format)
SFAUX output - Complex value of response

1. Independent s plane frequency to be used in evaluating the response is determined in subroutine FREQS1 or FREQS2.

### Requirements

COMMON blocks: FRQBLK,HEADDB,LENGTH
LCAP2 routines: none

## SFAUX1

### Identification

COMPLEX FUNCTION SFAUX1 - Evaluate S Plane Transfer Function Coefficient Array
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

This complex function is similar to SFAUX except that it is written so that it can be easily modified by the user to allow creation of a user-defined s plane transfer function.

### Usage

```
SFAUX1(TFC)
```

| | | |
|---|---|---|
| TFC | input | - Transfer function coefficient array (LCAP2 format) |
| SFAUX1 | output | - Complex value of response |

### Method

This complex function has only one line of code

```
SFAUX1=SFAUX(TFC)
```

so that it will yield the same results as SFAUX.

To create a user defined s plane transfer function, a different value is returned for SFAUX1. For example, if the function is

SPTF2 + SPTF4/2.

the user would change the FORTRAN code to

```
SFAUX1=SFAUX(SPTF2) + SFAUX(SPTF4)/2.
```

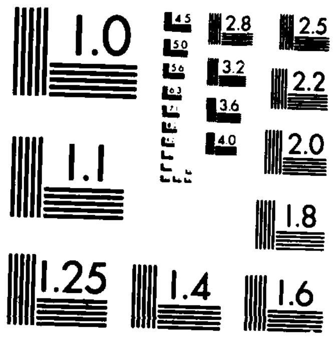
### Restrictions

The argument TFC must be an array in memory. Since only the first five s plane transfer functions are in COMMON/SCMBLK/ and all others are on a disk file, only SPTF1, SPTF2, SPTF3, SPTF4 and SPTF5 can be used to create a user-defined s plane transfer function. However, the user can define additional transfer function coefficient arrays in a separate labeled common block to be accessible by SFAUX. Subroutine FETSTF can be used to copy transfer functions from the disk file to the transfer function in this common block.

Requirements

COMMON blocks: FREQBLK,HEADDB,SCMBLK
LCAP2 routines: SFAUX

AD-A141 386 LCAP2 (LINEAR CONTROLS ANALYSIS PROGRAM) VOLUME 3 3/4
SOURCE CODE DESCRIPTION..(U) AEROSPACE CORP EL SEGUNDO
CA GUIDANCE AND CONTROL DIV E A LEE 15 NOV 83
UNCLASSIFIED TR-0084(9975)-1-VOL-3 SD-TR-84-06-VOL-3 F/G 9/2 NL




MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

## SFREQ

### Identification

SUBROUTINE SFREQ - LCAP2 Operator, S Plane Frequency Response
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Evaluate s plane frequency response using an LCAP2 index. Automatic frequency mode available to allow program to dynamically choose its own frequency points to yield a smooth plot of the response.

### Usage

```
CALL SFREQ(I)
```

I input - Index of s plane transfer function

1. Frequency response parameters are in COMMON/HEADDB. They are to be set before SFREQ is called. These parameters are defined below:

| parameter | preset | description |
|---|---|---|
| FAUTO | 1 | .NE.0 for automatic frequency mode. Uses NOMEG and OMEGA array.<br>.EQ.0 for user-supplied frequency points. Uses FREQ1, FREQ2, ..., FREQ5 arrays. |
| NOMEG | 2 | Number of frequency points entered in OMEGA array for use in auto. frequency mode (max=20) |
| OMEGA | | Array of frequency points for auto. freq. mode (Units defined by RAD). |
| | 1. | OMEGA(1)=first frequency point used in auto. mode |
| | 10. | OMEGA(2)=second frequency point used in auto. mode |
| | | OMEGA(NOMEG)=last freq. point used in auto. mode |
| RAD | 1 | .NE.0 for rad/sec, .EQ.0 for HZ |
| FBODE | 1 | .NE.0 for Bode plot |
| FNICO | 0 | .NE.0 for Nichols plot |
| PMARG | 0 | .NE.0 for plotting phase margin instead of phase for Nichols plot |
| FNYQS | 0 | .NE.0 for Nyquist plot |
| NQDB | 0 | .NE.0 for Nyquist plot in db |
| GRAFP | 1 | .NE.0 for printer (low resolution) plot |
| FILM | 0 | .NE.0 for hardcopy (high resolution) plot |
| FDLAY | 0 | Time delay (dead time) (s plane only) |
| DEGMN | -360. | Minimum defined phase in frequency response (Phase defined from DEGMN to DEGMN+360.) |

| | | |
|---|---|---|
| CYCLE | 0 | .EQ.0 for automatic selection of 2 or 3 cycle scale for Bode plots. (1 cycle not available) |
| FREQ1(1) | 1 | Starting freq. point for first segment of user specified values |
| FREQ1(2) | 10 | End freq. point for second segment of user specified values |
| FREQ1(3) | 1 | Delta frequency for third segment of user specified values |
| FREQk(1) | 0 | Starting freq. point for k-th segment * |
| FREQk(2) | 0 | End freq. point for k-th segment * |
| FREQk(3) | 0 | Delta frequency for k-th segment * |
| (k=2,5) | | (* - only if FAUTO.EQ.0) |
| DBMAX | 0 | Maximum db for plotting frequency response |
| DBMIN | 0 | Minimum db for plotting frequency response (Auto. scaling if DBMAX=DBMIN) |
| FXYDL | .5 | Nyquist plot scale in units per inch (Auto. scaling if FXYDL=0) |
| FXYMN | -2.5 | Nyquist plot parameter - minimum real and imaginary value plotted. |
| SAMPT | 1. | Sampling period for use in w or z plane freq. response |

2. Computed response variables are in blank common //. These variables are defined below:

| Variables | Description |
|---|---|
| NI | Number of plot points |
| XPOINT | Array of db of the response (DIMENSION=1500) |
| YPOINT | Array of phase of the response (DIMENSION=1500) |
| OMEGPT | Array of real frequencies (DIMENSION=1500) |
| CXR | Array of real part of response (DIMENSION=1500) |
| CXI | Array of imag part of response (DIMENSION=1500) |

Method

If the automatic frequency mode is selected (FAUTO=0), the program will choose frequency values for evaluating the transfer function such that successive delta db and delta phase values will be within specified limits to yield a smooth plot. The program evaluates the first point using f = OMEGA(1). Then choosing deltaf = OMEGA(1)/20 initially, the next frequency to be used is computed as f = f + deltaf. Evaluating the next point using this value of f, the delta db and delta phase is compared to the specified limits. If either is too large, deltaf is halved and the response is recomputed. If both are too small, deltaf is doubled and the response is recomputed. The limits for the delta db response is EDB1/2 and EDB1. The limits for the delta phase response is EDEG1/2 and EDEG1. Simultaneously with computing the next f to be used, a comparison is made with the next value of OMEGA(i). If f is larger than OMEGA(i), f will be replaced with the value of OMEGA(i). This will ensure that the user specified

frequency values will be inserted into those computed by the program. This process will continue until the last value OMEGA(NOMEG) is used.

Since the plot points computed to generate a smooth plot will, in many cases, be very large, only a portion of the computed response will be printed out. The printout is controlled by the delta db and delta phase parameters, EDB2 and EDEG2, respectively. A printout is made only if either of these two limits are exceeded.

Also as part of the automatic frequency mode, a comparison is made on deltaf to keep (deltaf/f) within the limits of MNDW and MXDW. The lower limit MNDW is necessary to prevent an excessive number of plot points around frequencies with low damping coefficients. The upper limit MXDW will ensure enough points to yield a smooth Bode plot.

The above parameters used in the automatic frequency mode are in COMMON/HEADDB/. They can be changed by the user. These parameters are defind below:

| parameter | preset | description |
|---|---|---|
| EDB1 | 1. | Min. delta db for plotting |
| EDB2 | 2. | Min. delta db for printout |
| EDEG1 | 4. | Max. delta phase for plotting |
| EDEG2 | 10. | Max. delta phase for printout |
| MNDW | .0005 | Min relative frequency step size |
| MXDW | .2 | Max. relative frequency step size |
| MXITF | 3000. | Max. no. of iterations in auto. mode |

With either the automatic or the non-automatic frequency mode, the program will automatically check for the gain and phase crossover. When found, interpolation is used to find the exact crossover frequency and the response computed at that frequency.

In the non-automatic frequency mode (FAUTO=0) the user can define up to five sets of frequencies to be used in computing the response. Each of these sets are specified by a three element array of the form FREQk(i), i=1,3 described above under Usage. If FREQk(1) = a, FREQk(2) = b, and FREQk(3) = c, the k-th set of frequencies specified is:

a, a+c, a+2c, ..., a+jc, b

where j is the largest integer such that (a+jc) is less than b. Each successive FREQk array must define an increasing set of frequencies such that the first value of the segment is always larger than the last value of the preceding segment. When FREQk(3) is not larger than FREQk(1), as in the case with the preset values for k = 2,5, those segments will not be used.

Requirements

COMMON blocks: FRQBLK,INTCOM,PRNCTL,TFTEMP,//
LCAP2 routines: ENDLINE,FCNPLN,FETTFX,FREQS2,FREQW2,FREQZ2,OPMESG,SFAUX,
WFAUX,ZFAUX

## SFREQY

### Identification

SUBROUTINE SFREQY - Evaluate Frequency Response Of An S Plane Transfer Function Coefficient Array
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Evaluate frequency response of an s plane transfer function coefficient array. User supplies name of the array.

### Usage

```
CALL SFREQY(TFC)
```

TFC input - Transfer function coefficient array (LCAP2 format)

1. Frequency response parameters are in COMMON/HEADDB/. See description of subroutine SFREQ for definition.

### Restrictions

If LCAP2 defined transfer function coefficient arrays are to be used, only the first five transfer functions for each plane are available since the others are on disk files. However, a user common block can be defined so that these other transfer functions can be first transferred from disk file to memory with subroutine FETSTF so that SFREQY can be used.

### Requirements

COMMON blocks: none
LCAP2 routines: FREQS2,SFAUX

# SLOCI

## Identification

SUBROUTINE SLOCI - LCAP2 Operator, S Plane Root Locus
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Evaluate s plane root locus using an LCAP2 index. Automatic gain selection is available to supplement user selected gains.

## Usage

```
CALL SLOCI(I)
```

I input - Index of s plane transfer function to be evaluated

1. Root locus parameters are in COMMON/HEADDB/. They are to be set before SLOCI is called. These parameters are defined below:

| parameter | preset | description |
|---|---|---|
| NLOCI | 2 | Number of root locus gains entered in array KGAIN (max=25) |
| KGAIN | | Array of values used for computing root locus gains |
| | .5 | KGAIN(1)=first user-specified root locus gain |
| | 2. | KGAIN(2)=second user-specified root locus gain |
| | | KGAIN(NLOCI)=last root locus gain (Gains computed and used only if they are between KGAIN(1) and KGAIN(NLOCI) ) |
| KFLG | 0 | .EQ.0 to increment gain by multiplying by KDELT<br>.NE.0 to increment gain by adding by KDELT |
| KDELT | 1.E4 | Value for changing gains (preset to large value so that no additional gains are computed by the program) |
| ITLOC | 50 | Max. no. of differenet gains computed for root locus |
| GRAFP | 1 | .NE.0 for printer (low resolution) plot |
| FILM | 0 | .NE.0 for hardcopy (high resolution) plot |
| RLXMN | 0 | Minimum x axis for plot |
| RLXMX | 0 | Maximum x axis for plot (Auto. scaling of x axis if RLXMN=RLXMX=0) |
| RLYMN | 0 | Minimum y axis for plot |
| RLYMX | 0 | Maximum y axis for plot (Auto. scaling of y axis if RLYMN=RLYMX=0) |

**Method**

Root locus is computed by evaluating the roots of the polynomial (PN + GAIN*PD) where GAIN is the varied gain and PN and PD are the numerator and denominator polynomials of the transfer function.

**Requirements**

COMMON blocks: INTCOM,PRNCTL,TFTEMP,//
LCAP2 routines: ENDLINE,FCNPLN,FETTFX,OPMESG,RLOCUS1,TFPRN1

## SNORM

### Identification

SUBROUTINE SNORM - LCAP2 Operator, Normalize S Plane Transfer Function
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Normalize s plane transfer function using an LCAP2 index. Normalization can be either with respect to the low order non-zero coefficient or the high order coefficient of the denominator.

### Usage

```
CALL SNORM(I)
```

I input - Index of the s plane transfer function

1. Normalization parameters are in COMMON/HEADDB/. They are to be set before SNORM is called. These parameters are defined below:

| parameter | preset | description |
|---|---|---|
| KNORM | 1. | Value used for normalizing the transfer function |
| NRMFG | 0 | If .EQ.0, the low order non-zero coefficient of the denominator is set equal to the value of KNORM and all other coefficients are normalized to this value. If KNORM=1., the low order non-zero coefficient of the numerator is the Bode gain. |
| | | If .NE.0, the high order coefficient of the denominator is set equal to the value of KNORM and all other coefficients are normalized to this value. If KNORM=1., the high order coefficient of the numerator is the root locus gain. |

### Restrictions

KNORM cannot be zero.

### Requirements

COMMON blocks: INTCOM,PRNCTL,TFTEMP
LCAP2 routines: ENDLINE,FCNPLN,FETTFX,NORM,OPMESG,STRTFX,TFPRN1,TFPRN4

## SPADD

### Identification

SUBROUTINE SPADD - LCAP2 Operator, S Plane Transfer Function Add
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Add two s plane transfer functions using LCAP2 indices.

### Usage

```
CALL SPADD(I,J,K)
```

I input - Index of resultant transfer function sum
J input - Index of first transfer function to be added
K input - Index of second transfer function to be added

### Restrictions

The degree of the transfer function must be less than 50.

### Requirements

COMMON blocks: PRNCTL,TFTEMP
LCAP2 routines: ENDLINE,FETTFX,OPPRN,PPADD,RRTADD,STFTFX,TFPRN1,TFPRN4

# SPDIV

## Identification

SUBROUTINE SPDIV - LCAP2 Operator, S Plane Transfer Function Divide
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Divide two s plane transfer functions using LCAP2 indices.

## Usage

```
CALL SPDIV(I,J,K)
```

I input - Index of resultant transfer function
J input - Index of dividend transfer function
K input - Index of divisor transfer function

## Restrictions

The degree of the transfer functions must be less than 50.

## Requirements

COMMON blocks: PRNCTL,TFTEMP
LCAP2 routines: ENDLINE,FETTFX,OPPRN,PMULT,PSYNTH,RTMPY,STRTFX,TFPRN1,
TFPRN4

## SPEQU

### Identification

SUBROUTINE SPEQU - LCAP2 Operator, S Plane Equal
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Equate s plane transfer functions using LCAP2 indices.

### Usage

```
CALL SPEQU(I,J)
```

I input - Index of resultant transfer function
J input - Index of transfer function to be equated with

### Restrictions

The degree of the transfer functions must be less than 50.

### Requirements

COMMON blocks: OVCOM,PRNCTL,TFTEMP
LCAP2 routines: ENDLINE,FETTFX,OPPRN,PPEQU,RRTEQU,STRTFX,TFPRN1,TFPRN4

## SPLDC

### Identification

SUBROUTINE SPLDC - LCAP2 Operator, Load Coefficients Into S Plane Transfer Function
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Load coefficients into s plane transfer function using an LCAP2 index.

### Usage

CALL SPLDC(I)

I input - Index where transfer function is to be stored

1. Transfer function coefficients are entered with polynomial coefficient arrays POLYN and POLYD (LCAP2 format) which are in COMMON/HEADDB/. They are to be set before SPLDC is called.

2. The calling program must include COMMON/HEADDB/ and the appropriate DIMENSION and EQUIVALENCE statements for POLYN and POLYD.

3. The roots of SPTFi will not be automatically computed. If this is desired, follow this operation with the operator SPRTS(I).

### Restrictions

The degree of the transfer function must be less than 50.

### Requirements

COMMON blocks: INTCOM,HEADDB,PRNCTL,TFTEMP
LCAP2 routines: ENDLINE,FCNPLN,OPMESG,PEQUAL,PPRN1,STRTFX,TFPRN4

# SPLDR

## Identification

SUBROUTINE SPLDR - LCAP2 Operator, Load S Plane Transfer Function In Root Form
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Load roots into s plane transfer function using an LCAP2 index.

## Usage

CALL SPLDR(I)

I input - Index where transfer function is to be stored

1. Transfer function roots are entered with polynomial coefficient arrays ROOTN and ROOTD (LCAP2 format) which are in COMMON/HEADDB/. They are to be set before SPLDR is called.

2. The calling program must include COMMON/HEADDB/ and the appropriate DIMENSION and EQUIVALENCE statements for ROOTN and ROOTD.

## Restrictions

The degree of the transfer function must be less than 50.

## Requirements

COMMON blocks: INTCOM,HEADDB,PRNCTL,TFTEMP
LCAP2 routines: ENDLINE,FCNPLN,OPMESG,PSYNTH,RTEQU,RTPRN2,STRTFX,TFPRN4

## SPMPY

### Identification

SUBROUTINE SPMPY - LCAP2 Operator, S Plane Transfer Function Multiply
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Multiply two s plane transfer functions using LCAP2 indices.

### Usage

```
CALL SPMPY(I,J,K)
```

I input - Index of resultant transfer function product
J input - Index of first transfer function to be multiplied
K input - Index of second transfer function to be multiplied

### Method

If only the coefficients of the j-th and k-th transfer functions are available, the product is computed by multiplication of the coefficients. If the roots of the j-th and k-th transfer functions are available, the product is computed by combining the roots. The coefficients of the product are then formed from these roots.

### Restrictions

The degree of the transfer function must be less that 50.

### Requirements

COMMON blocks: TFTEMP
LCAP2 routines: ENDLINE,FETTFX,OPPRN,PMULT,PSYNTH,RTMPY,STFTFX,TFPRN1,
TFPRN4

# SPPRN

## Identification

SUBROUTINE SPPRN - LCAP2 Operator, Print Out S Plane Transfer Function
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Print out s plane transfer function using an LCAP2 index.

## Usage

```
CALL SPPRN(I)
```

I  input  - Index of transfer function to be printed out

## Method

Roots of the transfer function are printed out only if they are defined (previously computed or loaded in). Coefficients of the transfer function are printed out in ascending order.

## Requirements

COMMON blocks: INTCOM,TFTEMP
LCAP2 routines: ENDLINE,FCNPLN,FETTFX,OPMESG,TFPRN4

# SPRTS

## Identification

SUBROUTINE SPRTS - LCAP2 Operator, Find Roots Of S Plane Transfer Function
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Find roots of an s plane transfer function using an LCAP2 index.

## Usage

```
CALL SPRTS(I)
```

I input - Index of s plane transfer function

## Method

Roots of the numerator and denominator are computed by subroutine PROOT.

## Restrictions

If the roots of SPTFi were previously computed or loaded in, the program will not recompute the roots from the coefficients. A message to this effect will be printed.

## Requirements

COMMON blocks: INTCOM,PRNCTL,TFTEMP
LCAP2 routines: ENDLINE,FCNPLN,FCNW1,FETTFX,OPMESG,PROOT,STRTFX,TFPRN1,
TFPRN4

# SPSUB

## Identification

SUBROUTINE SPSUB - LCAP2 Operator, S Plane Transfer Function Subtract
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Subtract two s plane transfer functions using LCAP2 indices.

## Usage

```
CALL SPSUB(I,J,K)
```

I input - Index of resultant transfer function difference
J input - Index of first transfer function (minuend)
K input - Index of second transfer function (subtrahend)

## Restrictions

The degree of the transfer functions must be less than 50.

## Requirements

COMMON blocks: PRNCTL,TFTEMP
LCAP2 routines: ENDLINE,FETTFX,OPPRN,PPSUB,RRTSUB,STRTFX,TFPRN1,TFPRN4

# STIME

## Identification

SUBROUTINE STIME - LCAP2 Operator, Inverse Laplace Transform And Time Response
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Compute inverse Laplace transform and the time response using an LCAP2 index.

## Usage

```
CALL STIME(I)
```

I input - Index of s plane transfer function

1. Time response parameters are in COMMON/HEADDB/. They are to be set before STIME is called. These parameters are defined below:

| parameter | preset | description |
|---|---|---|
| TSTEP | 1 | .NE.0 for step response; .EQ.0 for impulse response |
| TMAGN | 1. | Magnitude of input for time response |
| TZERO | 0 | Start time for evaluating time response |
| TEND | 1. | End time for evaluating time response |
| TDELT | 1. | Delta time for evaluating time response |
| TMAGN | 1. | Magnitude of input for time response |
| GRAFP | 1 | .NE.0 for printer (low resolution) plot |
| FILM | 0 | .NE.0 for hardcopy (high resolution) plot |
| TXMIN | 0 | Minimum x axis for plot |
| TXMAX | 0 | Maximum x axis for plot<br>(Auto. scaling of x axis if TXMIN=TXMAX) |
| TYMIN | 0 | Minimum y axis for plot |
| TYMAX | 0 | Mamimum y axis for plot<br>(Auto. scaling of y axis if TYMIN=TYMAX) |

## Method

The partial fraction expansion of the s plane transfer function times (1/s), if the input is a step function, is first computed. By utilizing the inverse Laplace transform, the analytical solution is computed and printed out. This analytical solution is then evaluated over the range of time values specified.

## Restrictions

The degree of the transfer function must be less than 50.

Due to the algorithm used to implement the partial fraction expansion, the following restrictions on the form of the s plane transfer function apply. Multiple poles are not allowed except for those at the origin. The poles at the origin (including the pole due to the 1/s term if the input is a step function) must be 5 or less. Also, the degree of the numerator must not be greater than the number of poles not at the origin.

## Requirements

COMMON blocks: INTCOM,PRNCTL,TFTEMP,//
LCAP2 routines: ENDLINE,FCNPLN,FETTFX,OPMESG,PROOT,STIME1,STFTFX,TFPRN1, ZTIME1

# STIME1

## Identification

SUBROUTINE STIME1 - Inverse Laplace Transform and Time Response
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Compute inverse Laplace transform and time response. This subroutine is called by STIME.

## Usage

```
CALL STIME1(PNS,PDS,RTNS,RTDS)
```

| | | |
|---|---|---|
| PNS | input | - Numerator polynomial coefficient array (LCAP2 format) |
| PDS | input | - Denominator polynomial coefficient array (LCAP2 format) |
| RTNS | input | - Complex numerator polynomial root array (LCAP2 format) |
| RTDS | input | - Complex denominator polynomial root array (LCAP2 format) |

1. Time response parameters are in COMMON/HEADDB/. See description of STIME.

## Method

Subroutine STIME2 is called to compute the analytical solution and STIME3 is called to evaluate the time response.

For interactive LCAP2 the user is prompted for the type of input and the beginning and end times to be used for evaluating the response. The user is also given the option to suppress tabular output of the response.

## Requirements

COMMON blocks: CMELIM,CMRESD,CMTIME,HEADDB,INTCOM,//
LCAP2 routines: GRAF1,ITITLE,STIME2,STIME3,TSPLOT,ZTIME2

## STIME2

### Identification

SUBROUTINE STIME2 - Compute Analytical Solution of Inverse Laplace Transform
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Compute analytical solution of inverse Laplace transform. This subroutine is called by STIME1.

### Usage

CALL STIME2(PNS,PDS,RTNS,RTDS,IPRN1,IPRN2,IPRN3)

| | | |
|---|---|---|
| PNS | input | - Numerator polynomial coefficient array (LCAP2 format) |
| PDS | input | - Denominator polynomial coefficient array (LCAP2 format) |
| RTNS | input | - Complex numerator polynomial root array (LCAP2 format) |
| RTDS | input | - Complex denominator polynomial root array (LCAP2 format) |
| IPRN1 | input | - .NE.0 for printout of common roots found |
| IPRN2 | input | - .NE.0 for printout of partial fraction coefficients |
| IPRN3 | input | - .NE.0 for printout of analytical solution |

1. Time response parameters are in COMMON/HEADDB/. See description for STIME.

2. Output is saved in COMMON/CMELIM/, COMMON/CMRESD/ and COMMON/CMTIME/ for use by subroutine STIME3. The parameters in these common blocks are described below:

| parameter (with dimension) | description |
|---|---|
| COMMON/CMELIM/ | |
| QROT(50) | Complex root array of the numerator (LCAP2 format) |
| PROT(50) | Complex root array of the denominator (LCAP2 format) |
| IQCOM | Number of complex roots of the numerator |
| IQR | Number of non-zero real roots of the numerator |
| IQZ | Number of roots at the origin of the numerator |
| IPCOM | Number of complex roots of the denominator |
| IPR | Number of non-zero real roots of the denominator |
| IPZ | Number of roots at the origin of the denominator |

COMMON/CMRESD/

| | |
|---|---|
| GAMMA(50) | Complex array of partial fraction coefficients of the poles not at the origin |
| LAMBDA(10) | Complex array of partial fraction coefficients of the poles at the origin origin |

COMMON/CMTIME/

| | |
|---|---|
| A(25) | Coefficient array for analytical expression of the response |
| B(25) | " |
| C(25) | " |
| ALPHA1(25) | " |
| ALPHA2(25) | " |
| BETA1(25) | " |
| BETA2(25) | " |

## Method

See description for STIME.

## Requirements

COMMON blocks: CMELIM,CMRESD,CMTIME,HEADDB
LCAP2 routines: CRELIM,RESDU,RTPRNO

## STIME3

### Identification

SUBROUTINE STIME3 - Compute S Plane Time Response From Analytical Solution
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Compute s plane time response from analytical solution computed by STIME2. This subroutine is called by STIME1.

### Usage

```
CALL STIME3(TPOINT,YPOINT,NI,IPRN1)
```

TPOINT output - Array of time points (DIMENSION=1500)
YPOINT output - Array of response points (DIMENSION=1500)
NI output - Number of response points
IPRN1 input - .NE.0 for tabular printout of time response

1. Time response parameters are in COMMON/HEADDB/. See description for STIME.

2. Parameters for the analytical solution as computed by subroutine STIME2 is in COMMON/CMELIM/, COMMON/CMRESD/ and COMMON/CMTIME/.

### Requirements

COMMON blocks: CMELIM,CMRESD,CMTIME,HEADDB
LCAP2 routines: LEXIT

# STORE

## Identification

SUBROUTINE STORE - Store Polynomial, Transfer Function and Matrix Data
CDC FORTRAN 4
C. L. Wong and E. A. Lee
Aerospace Corporation

## Purpose

Store data from an LCAP2 batch job for a restart capability. This data can be accessed in a subsequent batch or interactive job by using the RESTORE operator.

## Usage

```
CALL STORE(IPRNFG)
```

IPRNFG input - =0 for no printout of data stored

1. To identify the data stored, enter alphanumeric information in HEAD(64) through HEAD(70) of COMMON/HEADDB/ before calling STORE. This information will be printed out when this data is restored in a subsequent job.

## Method

Data will be saved on file TAPE31. The first record will be alphanumeric information copied from HEAD(64) through HEAD(70) of COMMON/HEADDB/. The second record will be information from COMMON/TFPCNT/ which describes the number of polynomials and transfer functions saved on various files. The third record will be LCAP2 parameters from HEAD(101) through HEAD(900) of COMMON/HEADDB/.

Polynomials and s, w and z plane transfer functions with LCAP2 indices 1 through 5 are stored in COMMON/SCMBLK/. These polynomials and transfer functions, regardless if they have been used by the user, will be the next data copied onto file TAPE31. Next, polynomials from file TAPE84 will be copied onto file TAPE31. Then s, w and z plane transfer functions from files TAPE85, TAPE86 and TAPE87, respectively, will be copied onto file TAPE31. Finally, matrix data from COMMON/MATRIX1/ and MDET1 will be copied onto file TAPE31.

## Restrictions

File type for TAPE31 must be declared with 'FILE,TAPE31,BT=I.'

## Requirements

COMMON blocks: ACOM,HEADDB,MATRIX1,MDET1,PRNCTL,SCMBLK,TFPCNT
LCAP2 routines: BPRINT1,ENDLINE,MPRINT1,OPMESG,PYPRN1,TFPRN1

## STRPY

### Identification

SUBROUTINE STRPY - Store polynomial
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Store polynomial in LCAP2 format.

### Usage

```
CALL STRPY(INDX,TFPOLY,TFROOT,IN)
```

| | | |
|---|---|---|
| INDX | input | - Index of polynomial to be stored |
| TFPOLY | input | - Polynomial coefficient array (LCAP2 format) |
| TFROOT | input | - Complex polynomial root array (LCAP2 format), only if IN=0 |
| IN | input | - =1 for coefficient form only<br>=0 for coefficient and root form |

### Method

If INDX .LE.5, polynomial data will be saved in COMMON/SCMBLK/. If INDX .GT.5 the polynomial data will be written on sequential file TAPE84. The counter NPYCNT in COMMON/TFPCNT/ is the number of polynomial records on sequential file TAPE84. This counter is incremented by one after the store is completed.

### Requirements

COMMON blocks: SCMBLK,TFPCNT
LCAP2 routines: none

## STRSTF

### Identification

SUBROUTINE STRSTF - Store S Plane Transfer Function
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Store s plane transfer function in LCAP2 format.

### Usage

```
CALL STRSTF(INDX,TFPOLY,TFROOT,IN,ID)
```

| | | | |
|---|---|---|---|
| INDX | input | - | Index of s plane transfer function |
| TFPOLY | input | - | Transfer function coefficient array (LCAP2 format) |
| TFROOT | input | - | Complex transfer function root array (LCAP2 format) |
| IN | input | - | =1 for numerator coefficient form only<br>=0 for numerator coefficient and root form |
| ID | input | - | =1 for denominator coefficient form only<br>=0 for denominator coefficient and root form |

### Method

If INDX .LE.5, transfer function data will be saved in COMMON/SCMBLK/. If INDX .GT.5 the transfer function data will be written on sequential file TAPE85. Counter NSPCNT in COMMON/HEADDB/ is used to keep track of the number of records written on file TAPE85. This counter is incremented by one after the store operation is completed.

### Requirements

COMMON blocks: SCMBLK,TFPCNT
LCAP2 routines: none

# STRTFX

## Identification

SUBROUTINE STRTFX - Store Transfer Function
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Store transfer function in LCAP2 format.

## Usage

CALL STRTFX(IPLANE,INDX,TFPOLY,TFROOT,IN,ID)

| | | | |
|---|---|---|---|
| IPLANE | input | - | =0 for s plane, =-1 for w plane, =1 for z plane |
| INDX | input | - | Index of transfer function |
| TFPOLY | input | - | Transfer function coefficient array (LCAP2 format) |
| TFROOT | input | - | Complex transfer function root array (LCAP2 format) |
| IN | input | - | =1 for numerator coefficient form only<br>=0 for numerator coefficient and root form |
| ID | input | - | =1 for denominator coefficient form only<br>=0 for denominator coefficient and root form |

## Method

If INDX .LE.5, transfer function data will be saved in COMMON/SCMBLK/. If INDX .GT.5 the transfer function data will be written on a sequential file. TAPE85, TAPE86 and TAPE87 are used, respectively for s, w and z data. Counters NSPCNT, NWPCNT and NZPCNT in COMMON/TFPCNT/ are used to keep track of records written on files TAPE85, TAPE86 and TAPE87, respectively. This counter is incremented by one after the store operation is completed.

## Requirements

COMMON blocks: none
LCAP2 routines: STRSTF,STRWTF,STRZTF

## STRWTF

### Identification

SUBROUTINE STRWTF - Store W Plane Transfer Function
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

Subroutine STRWTF is similar to subroutine STRSTF except that it is for the w plane instead of the s plane.

## STRZTF

### Identification

SUBROUTINE STRZTF - Store Z Plane Transfer Function
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

Subroutine STRZTF is similar to subroutine STRSTF except that it is for the z plane instead of the s plane.

## SUBP

### Identification

SUBROUTINE SUBP - Subtract Polynomials (Coefficient Form)
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Subtract coefficients of two polynomials

### Usage

CALL SUBP(A,B,C)

A input - Polynomial coefficient array (LCAP2 format) (minuend)
B input - Polynomial coefficient array (LCAP2 format) (subtrahend)
C output - Polynomial coefficient array (LCAP2 format)

1. EPAD1 (preset=1.E10) in COMMON/HEADDB/ is used to test for negligible coefficients.

### Method

Coefficients of polynomial B are subtracted from coefficients of polynomial A and stored into polynomial C. A test is then made to see if the highest order coefficient is smaller than all the other coefficients by 1/EPAD1. If it is, then it is considered to be negligible and is set to zero and the order of the polynomial reduced by one. This test is then repeated.

The code for this routine is in subroutine ADDP.

### Restrictions

The degree of the polynomials must be less than 50.

### Requirements

COMMON blocks: HEADDB
LCAP2 routines: PCHEK,PEQUAL,PZERO

## SWMRX

### Identification

SUBROUTINE SWMRX - LCAP2 Operator, S to W Plane Multirate Transform
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Compute multirate (slow input, fast output) s to w plane transform using LCAP2 indices. The zero order hold, if included, is at the slower input sampling rate. The ratio of the output/input sampling rates must be an integer. (note: the w is not the w' defined by the Tustin's bilinear rule)

### Usage

CALL SWMRX(I,J)

I input - Index of w plane transfer function
J input - Index of s plane transfer function

1. SWMRX parameters are in COMMON/HEADDB/. They are to be set before SWMRX is called. These parameters are defined below:

| parameter | preset | description |
|---|---|---|
| DELAY | 0 | Time delay, (enter negative value for time advance) |
| SAMPT | 1. | Sampling period of the faster output sampler |
| NTGER | 1 | Integer ratio of output/input sampling rates (or, input/output sampling periods) |
| ZOH | 1 | .NE.0 for inclusion of zero order hold at the input |

### Method

Partial fraction expansion of the s plane transfer function (including the 1/s from the zero order hold if there is one) is computed by subroutine RESDU. The w transform (at the faster output sampling rate) of each term of the expansion is then computed. Next, the terms of the expansion are summed and rationalized. Roots of this intermediate transfer function are then found. This result is multiplied by the discrete contribution of the zero order hold (which is at the slower input sampling rate) to yield the desired transform.

### Restrictions

The algorithm used for computing the partial fraction expansion requires the following constraints on the s plane transfer function. Multiple poles are not allowed except for those at the origin. The poles at the origin (including the 1/s from the zero order hold if there is one) must be 5 or less. Also, the

degree of the numerator must not be greater than the number of poles not at the origin.

<u>Requirements</u>

COMMON blocks: INTCOM,HEADDB,PRNCTL,TFTEMP
LCAP2 routines: ENDLINE,FETTFX,OPPRN,PPEQU,PROOT,RREQU,STRTFX,TFPRN1,
TFPRN4,WMRTRAN,WZTRANS

# SWXFM

## Identification

SUBROUTINE SWXFM - LCAP2 Operator, S to W Plane Transform
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Compute sampled-data transform from s to w plane using LCAP2 indices. (note: this w is not the w' defined by the Tustin's bilinear rule)

## Usage

```
CALL SWXFM(I,J)
```

I input - Index of w plane transfer function
J input - Index of s plane transfer function

1. SWXFM parameters are in COMMON/HEADDB/. They are to be set before SWXFM is called. These parameters are defined below:

| parameter | preset | description |
|---|---|---|
| DELAY | 0 | Time delay, (enter negative value for time advance) |
| SAMPT | 1. | Sampling period |
| ZOH | 1 | .NE.0 for inclusion of zero order hold |

## Method

Partial fraction expansion of the s plane transfer function (including the 1/s from the zero order hold if there is one) is computed by subroutine RESDU. The w transform of each term of the expansion is then computed. Next, the terms of the expansion are summed and rationalized. Roots of this intermediate transfer function are then found. This result is multiplied by the discrete contribution of the zero order hold to yield the desired transform.

## Restrictions

The algorithm used for computing the partial fraction expansion requires the following constraints on the s plane transfer function. Multiple poles are not allowed except for those at the origin. The poles at the origin (including the 1/s from the zero order hold if there is one) must be 5 or less. Also, the degree of the numerator must not be greater than the number of poles not at the origin.

## Requirements

COMMON blocks: HEADDB,PRNCTL,TFTEMP
LCAP2 routines: ENDLINE,FCNW1,FCNW2,FETTFX,OPPRN,PROOT,STRTFX,TFPRN1,
TFPRN4,WTRANS

## SZMRX

### Identification

SUBROUTINE SZMRX - LCAP2 Operator, S to Z Plane Multirate Transform
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Compute multirate (slow input, fast output) s to z plane transform using LCAP2 indices. The zero order hold, if included, is at the slower input sampling rate. The ratio of the output/input sampling rates must be an integer.

### Usage

```
CALL SZMRX(I,J)
```

I input - Index of z plane transfer function
J input - Index of s plane transfer function

1. SZMRX parameters are in COMMON/HEADDB/. They are to be set before SZMRX is called. These parameters are defined below:

| parameter | preset | description |
|---|---|---|
| DELAY | 0 | Time delay, (enter negative value for time advance) |
| SAMPT | 1. | Sampling period of the faster output sampler |
| NTGER | 1 | Integer ratio of output/input sampling rates (or, input/output sampling periods) |
| ZOH | 1 | .NE.0 for inclusion of zero order hold at the input |

### Method

Partial fraction expansion of the s plane transfer function (including the 1/s from the zero order hold if there is one) is computed by subroutine RESDU. Since calculations performed in the w plane are more accurate than those in the z plane, the w transform (at the faster output sampling rate) of each term of the expansion is then computed. Next, the terms of the expansion are summed and rationalized. Roots of this intermediate transfer function are then found. This result is multiplied by the discrete contribution of the zero order hold (which is at the slower input sampling rate) to yield the w plane form of the desired transform. Next, subroutine WZTRANS is called to transform the roots w plane roots to the z plane. The coefficients of the z plane transfer function are then computed.

The code for this routine is in subroutine SWMRX.

## Restrictions

The algorithm used for computing the partial fraction expansion requires the following constraints on the s plane transfer function. Multiple poles are not allowed except for those at the origin. The poles at the origin (including the 1/s from the zero order hold if there is one) must be 5 or less. Also, the degree of the numerator must not be greater than the number of poles not at the origin.

## Requirements

COMMON blocks: INTCOM,HEADDB,PRNCTL,TFTEMP
LCAP2 routines: ENDLINE,FETTFX,OPPRN,PPEQU,PROOT,RREQU,STRTFX,TFPRN1, TFPRN4,WMRTRAN,WZTRANS

## SZXFM

### Identification

SUBROUTINE SZXFM - LCAP2 Operator, S To Z Plane Transform
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Compute sampled-data transform from s to z plane using LCAP2 indices.

### Usage

```
CALL SZXFM(I,J)
```

I input - Index of z plane transfer function
J input - Index of s plane transfer function

1. SZXFM parameters are in COMMON/HEADDB/. See description for SWXFM.

### Method

Partial fraction expansion of s plane transfer function (including the 1/s from the zero order hold if there is one) is computed by subroutine RESDU. Since calculations performed in the w plane are more accurate than those in the z plane, the w transform of each term of the partial fraction expansion is computed. Next, the terms of the expansion are summed and rationalized. Roots of this intermediate transfer function are then found. This result is multiplied by the discrete contribution of the zero order hold to yield the w plane form of the desired transform. Next, subroutine WZTRANS is called to transform the w plane roots to the z plane. The coefficients of the z plane transfer function are then computed.

### Restrictions

Same restrictions on s plane transfer function that apply for SWXFM. For higher order s plane transfer functions, the w plane transform will be more accurate than the z plane transform. To determine differences in numerical accuracies, compute both w and z plane transforms and compare frequency responses.

### Requirements

COMMON blocks: HEADDB,PRNCTL,TFTEMP
LCAP2 routines: ENDLINE,FCNPLN,FCNW1,FCNW2,FETTFX,HOLLI,OPPRN,PROOT,
STRTFX,TFPRN1,TFPRN4,WTRANS,WZTRANS

## TFAUX

### Identification

COMPLEX FUNCTION TFAUX - Transfer Function Evaluation
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Evaluate transfer function coefficient array (LCAP2 format). The frequency value to be used is specified explicitly as an argument.

### Usage

TFAUX(TFC,X)

| | | |
|---|---|---|
| TFC | input | - Transfer function coefficient array (LCAP2 format) |
| X | input | - Complex frequency used to evaluate the transfer function |
| TFAUX | output | - Complex value of the response |

### Restrictions

The degree of the transfer function must be less than 50.

### Requirements

COMMON blocks: none
LCAP2 routines: none

# TFPRN1

## Identification

SUBROUTINE TFPRN1 - Print Out Transfer Function
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Print out transfer function coefficients and roots under control of PRNFLG1.

## Usage

```
CALL TFPRN1(IPLANE,I,TFPI,TFRI,IN,ID)
```

```
IPLANE  input - =1HS for s plane, =1HW for w plane, =1HZ for z plane
I       input - Identifier (.GT.0) used for labeling printout
TFPI    input - Transfer function coefficient array (LCAP2 format)
TFRI    input - Complex transfer function root array (LCAP2 format)
IN      input - =1 for numerator coefficient form only
                =0 for numerator coefficient and root form
ID      input - =1 for denominator coefficient form only
                =0 for denominator coefficient and root form
```

1. Flag PRNFLG1 (preset=1) in COMMON/PRNCTL/ if .EQ.0 will suppress the printout.

## Requirements

COMMON blocks: PRNCTL
LCAP2 routines: PPRN1,RTPRN2

# TFPRN4

## Identification

SUBROUTINE TFPRN4 - Print Out Transfer Function
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Print out transfer function coefficients and roots under control of PRNFLG4.

## Usage

```
CALL TFPRN4(IPLANE,I,TFPI,TFRI,IN,ID)
```

```
IPLANE  input - =1HS for s plane, =1HW for w plane, =1HZ for z plane
I       input - Identifier (.GT.0) used for labeling printout
TFPI    input - Transfer function coefficient array (LCAP2 format)
TFRI    input - Complex transfer function root array (LCAP2 format)
IN      input - =1 for numerator coefficient form only
                =0 for numerator coefficient and root form
ID      input - =1 for denominator coefficient form only
                =0 for denominator coefficient and root form
```

1. Flag PRNFLG4 (preset=1) in COMMON/PRNCTL/ if .EQ.0 will suppress the print-out.

## Requirements

COMMON blocks: PRNCTL
LCAP2 routines: PPRN1,RTPRN2

# TSPLOT

## Identification

SUBROUTINE TSPLOT - Time Response Plot
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Plot time response computed by either STIME or ZTIME. This subroutine is called by STIME1 or ZTIME1.

## Usage

```
CALL TSPLOT(NI,TPOINT,YPOINT)
```

NI     input - Number of plot points in arrays TPOINT and YPOINT
TPOINT input - Array of time points (DIMENSION=1500)
YPOINT input - Array of time response points (DIMENSION=1500)

1. Time response plot parameters are in COMMON/HEADDB/. See description of STIME.

## Requirements

COMMON blocks: AWORDS,HEADDB,PLOT1
LCAP2 routines: ELPLOT1,OSCALE

# USRNOTE

## Identification

SUBROUTINE USRNOTE - User Note For Additional Labeling Of Nichols Plot
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

This is a dummy subroutine to satisfy the loader if the user does supply his/her own code for this routine. USRNOTE is intended to be used for additional annotation on the right portion of the Nichols plot.

## Usage

```
CALL USRNOTE
```

1. Code has been written for specific Aerospace studies to provide additional annotation of the Nichols plot. An example will be given in a future revision of this report.

## Requirements

COMMON blocks: none
LCAP2 routines: none

# WELCR

## Identification

SUBROUTINE WELCR - LCAP2 Operator, Eliminate Common Roots Of W Plane
Transfer Function
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Eliminate common roots from a w plane transfer function using an LCAP2 index.

## Usage

CALL WELCR(I)

I input - Index of w plane transfer function

1. Common root elimination parameters ECRE1 (preset=2.E-4) and ECRE2 (preset=1.E-8) are in COMMON/HEADDB/.

## Method

If a numerator root nrt and a denominator root drt are found such that ABS(drt/nrt - (1.,0.)).LT.ECRE1 for nrt.NE.0 or ABS(drt).LT.ECRE2 for nrt.EQ.0, roots nrt and drt are considered to be common roots and will be eliminated from the transfer function.

The code for this routine is in subroutine SELCR.

## Requirements

COMMON blocks: INTCOM,PRNCTL,TFTEMP
LCAP2 routines: CRELIM,ENDLINE,FCNPLN,FETTFX,OPMESG,OPPRN,PROOT,PSYNTH,
RREQU,RTPRNO,STRTFX,TFPRN1,TFPRN4

## WFAUX

### Identification

COMPLEX FUNCTION WFAUX - Evaluate W Plane Transfer Function Coefficient Array
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Evaluate w plane transfer function coefficient array (LCAP2 format) for use in computing the frequency response. This complex function can be used by subroutine FREQW1 or FREQW2 to evaluate the transfer function specified by is first argument. It can also be used by user-supplied subroutines similar to WFAUX1.

This subroutine can also evaluate the multirate (fast input, slow output) response of the transfer function.

### Usage

WFAUX(TFC)

TFC input - Transfer function coefficient array (LCAP2 format)
WFAUX output - Complex value of response

1. Independent w plane frequency used in evaluation of the response is computed by the program using real frequency X of COMMON/FRQBLK/ and sampling period SAMPT of COMMON/HEADDB/.

2. If MMTGER of COMMON/FRQBLK/ is .GT.0, the multirate response is computed by using Sklansky's frequency decomposition method. MMTGER is the ratio of the (output/input) sampling periods and SAMPT is the sampling period of the faster input sampler.

### Method

The code for this routine is in subroutine ZFAUX.

### Requirements

COMMON blocks: FRQBLK,HEADDB,LENGTH
LCAP2 routines: none

# WFAUX1

## Identification

COMPLEX FUNCTION WFAUX1 - Evaluate W Plane Transfer Function Coefficient Array
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

This complex function is similar to WFAUX except that it is written so that it can be easily modified by the user to allow creation of a user-defined w plane transfer function.

## Usage

```
WFAUX1(TFC)
```

TFC    input  - Transfer function coefficient array (LCAP2 format)
WFAUX1 output - Complex value of the response

## Method

This complex function has only one line of code

```
WFAUX1=WFAUX(TFC)
```

so that it will yield the same results as WFAUX.

To create a user-defined w plane transfer function, a different value is returned for WFAUX1. For example, if the function is

```
WPTF2 + WPTF4/2.
```

the user would change the FORTRAN code to

```
WFAUX1=WFAUX(WPTF2) + WFAUX(WPTF4)/2.
```

## Restrictions

The argument TFC must be an array in memory. Since only the first five w plane transfer functions are in COMMON/SCMBLK/ and all others are on a disk file, only WPTF1, WPTF2, WPTF3, WPTF4 and WPTF5 can be used to create a user-defined w plane transfer function. However, the user can define additional transfer function coefficient arrays in a separate labeled common block to be accessible by WFAUX. Subroutine FETWTF can be used to copy transfer functions from the disk file to the transfer function in this common block.

## Requirements

COMMON blocks: FRQBLK,HEADDB,SCMBLK
LCAP2 routines: CFUNC,WFAUX

# WFREQ

## Identification

SUBROUTINE WFREQ - LCAP2 Operator, W Plane Frequency Response
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Evaluate w plane frequency response using an LCAP2 index. Automatic frequency mode available to allow program to dynamically choose its own frequency points to yield a smooth plot of the response.

## Usage

CALL WFREQ(I)

I input - Index of w plane transfer function

1. Frequency response parameters are in COMMON/HEADDB. They are to be set before WFREQ is called. See description of SFREQ for the complete list of definitions of these parameters. The parameter SAMPT is described below:

| parameter | preset | description |
|---|---|---|
| SAMPT | 1 | Sampling period |

## Method

Same as that described in detail in description of SFREQ.

In the automatic frequency mode (FAUTO=0) the program will avoid using frequency values at or near the half sampling frequency. W plane frequencies (imaginary part) greater than 1000. will not be used.

The code for this routine is in subroutine SFREQ.

## Requirements

COMMON blocks: FRQBLK,INTCOM,PRNCTL,TFTEMP,//
LCAP2 routines: ENDLINE,FCNPLN,FETTFX,FREQS2,FREQW2,FREQZ2,OPMESG,SFAUX,
WFAUX,ZFAUX

# WFREQY

## Identification

SUBROUTINE WFREQY - Evaluate Frequency Response Of A W Plane Transfer
Function Coefficient Array
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Evaluate frequency response of a w plane transfer function coefficient array. User supplies name of the array.

## Usage

```
CALL WFREQY(TFC)
```

TFC input - Transfer function coefficient array (LCAP2 format)

1. Frequency response parameters are in COMMON/HEADDB/. See description of subroutine SFREQ for definition.

## Restrictions

If LCAP2 defined transfer function coefficient arrays are to be used, only the first five transfer functions for each plane are available since the others are on disk files. However, a user common block can be defined so that these other transfer functions can be first transferred from disk file to memory so that WFREQY can be used.

## Requirements

COMMON blocks: FRQBLK
LCAP2 routines: FREQW2,WFAUX

# WLOCI

## Identification

SUBROUTINE WLOCI - LCAP2 Operator, W Plane Root Locus
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Evaluate w plane root locus using an LCAP2 index. Automatic gain selection available to supplement user selected gains.

## Usage

```
CALL WLOCI(I)
```

I input - Index of w plane transfer function to be evaluated

1. Root locus parameters are in COMMON/HEADDB/. They are to be set before WLOCI is called. See description of SLOCI for a complete list of definition of these parameters.

## Method

Same as that described in detail in description of SLOCI.

The code for this routine is in subroutine SLOCI.

## Requirements

COMMON blocks: INTCOM,PRNCTL,TFTEMP,//
LCAP2 routines: ENDLINE,FCNPLN,FETTFX,OPMESG,RLOCUS1,TFPRN1

# WMFAUX

## Identification

COMPLEX FUNCTION WMFAUX1 - Evaluate Multirate W Plane Transfer Function
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Evaluate multirate (fast input, slow output) response of a w plane transfer function in coefficient form for use in computing the frequency response. This complex function can be used by subroutines FREQWM1 or FREQWM2 to evaluate the transfer function specified by its first argument.

## Usage

WMFAUX(TFC,M,T)

| | | |
|---|---|---|
| TFC | input | - Transfer function coefficient array (LCAP2 format) |
| M | input | - Integer ratio of output/input sampling periods |
| T | input | - Sampling period of slower outout sampler |
| WMFAUX1 | output | - Complex value of response |

## Method

Sklansky's frequency decomposition method is used to evaluate the transfer function.

## Requirements

COMMON blocks: FRQBLK,HEADDB
LCAP2 routines: WFAUX

# WMRFQ

## Identification

SUBROUTINE WMRFQ - LCAP2 Operator, W Plane Multirate Frequency Response
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Evaluate multirate (fast input, slow output) frequency response of a w plane transfer function using an LCAP2 index.

## Usage

```
CALL WMRFQ(I,M)
```

I input - Index of w plane transfer function
M input - Integer ratio of output/input sampling periods

1. The input w plane transfer function is at the faster sampling rate.

2. Frequency response parameters are in COMMON/HEADDB/. See description of SFREQ.

3. The sampling period, SAMPT, is for the slower output sampler.

## Method

The frequency response is evaluated by direct application of Sklansky's frequency decomposition. No explicit rational representation of the slower output transform is computed. If an explicit representation of the slower output transfer function is desired, see LCAP2 operator WMRXFM.

## Requirements

COMMON blocks: INTCOM,FRQBLK,PRNCTL,TFTEMP
LCAP2 routines: ENDLINE,FCNPLN,FCNW1,FETTFX,FREQWM2,FREQZM2,OPMESG,
TFPRN1,WFAUX,ZFAUX

# WMRFRQY

## Identification

SUBROUTINE WMRFQY - Evaluate Multirate Frequency Response Of A W Plane Transfer Function Coefficient Array
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Evaluate multirate (fast input, slow output) frequency response of a w plane transfer function in coefficient form. User supplies the name of the array.

## Usage

```
CALL WMRFQY(TFC,M)
```

| | | |
|---|---|---|
| TFC | input | - Transfer function coefficient array (LCAP2 format) at the faster input sampling rate |
| M | input | - Integer ratio of output/input sampling periods |

1. Frequency response parameters are in COMMON/HEADDB/. See description of of SFREQ.

2. The sampling period, SAMPT, is for the slower output sampler.

## Method

If LCAP2 defined transfer function coefficient arrays are to be used, only the first five transfer functions for each plane are available since the others are on disk files. However, a user common block can be defined so that these other transfer functions can be first transferred from disk file to memory so that WMRFRQY can be used.

## Requirements

COMMON blocks: FRQBLK
LCAP2 routines: FREQWM2,WFAUX

## WMRTRAN

### Identification

SUBROUTINE WMRTRAN - Compute S To W Plane Transformation
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Compute multirate (slow input, fast output) s to w plane transformation. This subroutine is called by SWMRX.

### Usage

```
CALL WMRTRAN(PNS,PDS,RTNS,RTDS,PNW,PDW,RTNW,RTDW,T,DELAY,IZOH)
```

| | | |
|---|---|---|
| PNS | input | - S plane numerator poly. coefficient array (LCAP2 format) |
| PDS | input | - S plane denominator poly. coefficient array (LCAP2 format) |
| RTNS | input | - S plane numerator poly. root array (LCAP2 format) |
| RTDS | input | - S plane denominator poly. root array (LCAP2 format) |
| PNW | output | - W plane numerator poly. coefficient array (LCAP2 format) |
| PDW | output | - W plane denominator poly. coefficient array (LCAP2 format) |
| RTNW | output | - W plane numerator poly. root array (LCAP2 format) |
| RTDW | output | - W plane denominator poly. root array (LCAP2 format) |
| T | input | - Sampling period |
| DELAY | input | - Time delay |
| IZOH | input | - .NE.0 for inclusion of zero order hold |

1. NTGER, integer ratio of input/output sampling periods, of COMMON/HEADDB/ must be set before WMRTRAN is called.

### Method

See description of SWMRX.

The code for this routine is in subroutine WTRANS.

### Restrictions

See description of SWMRX.

### Requirements

COMMON blocks: HEADDB,INTCOM,PRNCTL,//
LCAP2 routines: CRELIM,PEQUAL,PMULT,PPADD,PPRINT,PPZERO,PROOT,PSYNTH,
RESDU,RPRINT,RREQU,RRTAD1,RRTPRN,RRZERO,RTADD,RTPRNO,
RZERO,TFRPRN

## WMRXFM

### Identification

SUBROUTINE WMRXFM - LCAP2 Operator, Multirate (fast input, slow output sampler) W Transform

CDC FORTRAN 4

E. A. Lee

Aerospace Corporation

### Purpose

Compute the output w transform of a fast to slow sampler using LCAP2 indices. This operation will yield a rational transfer function at the slower sampling rate.

### Usage

CALL WMRXFM(I,J)

I input - Index of resultant slower output w plane transfer function

J input - Index of faster input w plane transfer function

1. The integer ratio, output/input sampling periods, NTGER, of COMMON/HEADDB/ must be set before this subroutine is called.

### Method

The faster input transfer function is first transformed to the z plane. The output transfer function of a fast to slow rate sampler is then given in the z plane by Sklansky's frequency decomposition method as,

$$\frac{1}{n}\sum_{k=1}^{n-1} G^{T/n}\left(z_n\, e^{j2*pi*k/n}\right)$$

where $G^{T/n}(z_n)$ is the z plane transfer function at the faster sampling rate and n is the integer ratio of the output/input sampling periods. Using the root form representation of the input transform, a rational representation of the slower rate output transfer function is computed. This transfer function is then transformed to the w plane.

The code for this routine is in subroutine ZMRXFM.

Requirements

COMMON blocks: HEADDB,ITEST,PRNCTL,TFTEMP
LCAP2 routines: CRELIM,ENDLINE,FETTFX,MRXFM,OPPRN,PROOT,PSYNTH,RREQU,
RRTEQU,RTPRNO,STRTFX,TFPRN1,TFPRN4,WZTRANS,ZWTRANS

# WNORM

## Identification

SUBROUTINE WNORM - LCAP2 Operator, Normalize W Plane Transfer Function
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Normalize w plane transfer function using an LCAP2 index. Normalization can be either with respect to the low order non-zero coefficient or the high order coefficient of the denominator.

## Usage

```
CALL WNORM(I)
```

I input - Index of the w plane transfer function

1. Normalization parameters are in COMMON/HEADDB/. They are to be set before WNORM is called. These parameters are defined below:

| parameter | preset | description |
|---|---|---|
| KNORM | 1. | Value used for normalizing the transfer function |
| KNRMFG | 0 | If .EQ.0, the low order non-zero coefficient of the denominator is set equal to the value of KNORM and all other coefficients are normalized to this value. If KNORM=1., the low order non-zero coefficient of the numerator is the Bode gain. If .NE.0, the high order coefficient of the denominator is set equal to the value of KNORM and all other coefficients are normalized to this value. If KNORM=1., the high order coefficient of the numerator is the root locus gain. |

## Method

The code for this routine is in subroutine SNORM.

## Restrictions

KNORM cannot be zero.

## Requirements

COMMON blocks: INTCOM,PRNCTL,TFTEMP
LCAP2 routines: ENDLINE,FCNPLN,FETTFX,NORM,OPMESG,STRTFX,TFPRN1,TFPRN4

# WPADD

## Identification

SUBROUTINE WPADD - LCAP2 Operator, W Plane Transfer Function Add
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Add two w plane transfer functions using LCAP2 indices.

## Usage

```
CALL WPADD(I,J,K)
```

I input - Index of resultant transfer function sum
J input - Index of first transfer function to be added
K input - Index of second transfer function to be added

## Method

The code for this routine is in subroutine SPADD.

## Restrictions

The degree of the transfer functions must be less than 50.

## Requirements

COMMON blocks: PRNCTL,TFTEMP
LCAP2 routines: ENDLINE,FETTFX,OPPRN,PPADD,RRTADD,STRTFX,TFPRN1,TFPRN4

# WPDIV

## Identification

SUBROUTINE WPDIV - LCAP2 Operator, W Plane Transfer Function Divide
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Divide two w plane transfer functions using LCAP2 indices.

## Usage

CALL WPDIV(I,J,K)

I input - Index of resultant transfer function
J input - Index of dividend transfer function
K input - Index of divisor transfer function

## Method

The code for this routine is in subroutine SPDIV.

## Restrictions

The degree of the transfer functions must be less than 50.

## Requirements

COMMON blocks: PRNCTL,TFTEMP
LCAP2 routines: ENDLINE,FETTFX,OPPRN,PMULT,PSYNTH,RTMPY,STRTFX,TFPRN1,
TFPRN4

## WPEQU

### Identification

SUBROUTINE WPEQU - LCAP2 Operator, W Plane Equal
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Equate w plane transfer functions using LCAP2 indices.

### Usage

CALL WPEQU(I,J)

I input - Index of resultant transfer function
J input - Index of transfer function to be equated with

### Method

The code for this routine is in subroutine SPEQU.

### Restrictions

*The degree of the transfer functions must be less than 50.*

### Requirements

COMMON blocks: OVCOM,PRNCTL,TFTEMP
LCAP2 routines: ENDLINE,FETTFX,OPPRN,PPEQU,RRTEQU,STRTFX,TFPRN1,TFPRN4

# WPLDC

## Identification

SUBROUTINE WPLDC - LCAP2 Operator, Load Coefficients Into W Plane Transfer Function
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Load coefficients into w plane transfer function using an LCAP2 index.

## Usage

CALL WPLDC(I)

I input - Index where transfer function is to be stored

1. Transfer function coefficients are entered with polynomial coefficient arrays POLYN and POLYD (LCAP2 format) which are in COMMON/HEADDB/. They are to be set before WPLDC is called.

2. The calling program must include COMMON/HEADDB/ and the appropriate DIMENSION and EQUIVALENCE statements for POLYN and POLYD.

3. The roots of WPTFi will not be automatically computed. If this is desired, follow this operation with the operator WPRTS(I).

## Method

The code for this routine is in subroutine SPLDC.

## Restrictions

The degree of the transfer function must be less than 50.

## Requirements

COMMON blocks: INTCOM,HEADDB,PRNCTL,TFTEMP
LCAP2 routines: ENDLINE,FCNPLN,OPMESG,PEQUAL,PPRN1,STRTFX,TFPRN4

# WPLDR

## Identification

SUBROUTINE WPLDR - LCAP2 Operator, Load W Plane Transfer Function In Root Form
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Load roots into w plane transfer function using an LCAP2 index.

## Usage

CALL WPLDR(I)

I input - Index where transfer function is to be stored

1. Transfer function roots are entered with polynomial coefficient arrays ROOTN and ROOTD (LCAP2 format) which are in COMMON/HEADDB/. They are to be set before WPLDR is called.

2. The calling program must include COMMON/HEADDB/ and the appropriate DIMENSION and EQUIVALENCE statements for ROOTN and ROOTD.

## Method

The code for this routine is in subroutine SPLDR.

## Restrictions

The degree of the transfer function must be less than 50.

## Requirements

COMMON blocks: INTCOM,HEADDB,PRNCTL,TFTEMP
LCAP2 routines: ENDLINE,FCNPLN,OPMESG,PSYNTH,RTEQU,RTPRN2,STRTFX,TFPRN4

## WPMPY

### Identification

SUBROUTINE WPMPY - LCAP2 Operator, W Plane Transfer Function Multiply
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Multiply two w plane transfer functions using LCAP2 indices.

### Usage

```
CALL WPMPY(I,J,K)
```

I input - Index of resultant transfer function product
J input - Index of first transfer function to be multiplied
K input - Index of second transfer function to be multiplied

### Method

If only the coefficients of the j-th and k-th transfer functions are available, the product is computed by multiplication of the coefficients. If the roots of the j-th and k-th transfer functions are available, the product is computed by combining the roots. The coefficients of the product are then formed from these roots.

The code for this routine is in subroutine SPMPY.

### Restrictions

The degree of the transfer function must be less that 50.

### Requirements

COMMON blocks: PRNCTL,TFTEMP
LCAP2 routines: ENDLINE,FCNW1,FCNW2,FETTFX,OPPRN,PMULT,PSYNTH,RTMPY,
STRTFX,TFPRN1,TFPRN4

# WPPRN

## Identification

SUBROUTINE WPPRN - LCAP2 Operator, Print Out W Plane Transfer Function
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Print out w plane transfer function using an LCAP2 index.

## Usage

CALL WPPRN(I)

I input - Index of transfer function to be printed out

## Method

Roots of the transfer function are printed out only if they are defined (previously computed or loaded in). The coefficients of the transfer function are printed out in ascending order.

The code for this routine is in subroutine SPPRN.

## Requirements

COMMON blocks: INTCOM,TFTEMP
LCAP2 routines: ENDLINE,FCNPLN,FETTFX,OPMESG,TFPRN4

# WPRTS

## Identification

SUBROUTINE WPRTS - LCAP2 Operator, Find Roots Of W Plane Transfer Function
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Find roots of a w plane transfer function using an LCAP2 index.

## Usage

```
CALL WPRTS(I)
```

I input - Index of w plane transfer function

## Method

Roots of the numerator and denominator are computed by subroutine PROOT.

The code for this routine is in subroutine SPRTS.

## Restrictions

If the roots of WPTFi were previously computed or loaded in, the program will not recompute the roots from the coefficients. A message to this effect will be printed.

## Requirements

COMMON blocks: INTCOM,PRNCTL,TFTEMP
LCAP2 routines: ENDLINE,FCNPLN,FCNW1,FETTFX,OPMESG,PROOT,STRTFX,TFPRN1,
TFPRN4

# WPSUB

## Identification

SUBROUTINE WPSUB - LCAP2 Operator, W Plane Transfer Function Subtract
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Subtract two w plane transfer functions using LCAP2 indices.

## Usage

```
CALL WPSUB(I,J,K)
```

I input - Index of resultant transfer function difference
J input - Index of first transfer function (minuend)
K input - Index of second transfer function (subtrahend)

## Method

The code for this routine is in subroutine SPSUB.

## Restrictions

The degree of the transfer functions must be less than 50.

## Requirements

COMMON blocks: PRNCTL,TFTEMP
LCAP2 routines: ENDLINE,FETTFX,OPPRN,PPSUB,RRTSUB,STRTFX,TFPRN1,TFPRN4

## WSTRAN1

### Identification

SUBROUTINE WSTRAN1 - Transform W Plane Roots Into S Plane
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Transform w plane roots into "equivalent" s plane roots. The transformation of the w plane roots to the s plane is not unique. The "equivalent" s plane roots are provided solely to aid the analyst in identifying and correlating w plane roots. The computed s plane roots are not saved. This subroutine is called by WSXFM.

### Usage

```
CALL WSTRAN1(IPLANE,WPTF,WROOT,SAMPT)
```

| | | |
|---|---|---|
| IPLANE | input | - .LT.0 for w plane, .GT.0 for z plane |
| WPTF | input | - W plane transfer function coefficient array (LCAP2 format) |
| WROOT | input | - W plane transfer function root array (LCAP2 format) |
| SAMPT | input | - Sampling period |

### Method

See description for WSXFM

### Requirements

COMMON blocks: none
LCAP2 routines: RTPRNO

# WSXFM

## Identification

SUBROUTINE WSXFM - LCAP2 Operator, Transform W Plane Roots Into S Plane
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Transform w plane roots into "equivalent" s plane roots using an LCAP2 index. The transformation of the w plane roots to the s plane is not unique. The "equivalent" s plane roots are provided solely to aid the analyst in identifying and correlating w plane roots. The computed s plane roots are not saved.

## Usage

```
CALL WSXFM(I)
```

I input - Index of w plane transfer function

1. Sampling period, SAMPT, of COMMON/HEADDB/ must be set before calling this subroutine.

## Method

Transformation of the roots from w to the s plane is defined by

```
s = ln( (1+w)/(1-w) ) / SAMPT
```

When w = -1.0 or +1.0 the "equivalent" s plane root is undefined. If ABS(1.-w) is less than 1.E-5, the equivalent root is printed out as 999999.99. If ABS(w+1.) is less than 1.E-5, the equivalent root is printed out as -999999.99.

## Requirements

COMMON blocks: INTCOM,HEADDB,PRNCTL,TFTEMP
LCAP2 routines: ENDLINE,FCNW2,FETTFX,OPMESG,TFPRN1,WSTRAN1

# WTRANS

## Identification

SUBROUTINE WTRANS - Compute S To W Plane Transformation
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Compute s to w plane transformation. This subroutine is called by SWXFM.

## Usage

```
CALL WTRANS(PNS,PDS,RTNS,RTDS,PNW,PDW,RTNW,RTDW,T,DELAY,IZOH)
```

| | | |
|---|---|---|
| PNS | input | - S plane numerator poly. coefficient array (LCAP2 format) |
| PDS | input | - S plane denominator poly. coefficient array (LCAP2 format) |
| RTNS | input | - S plane numerator poly. root array (LCAP2 format) |
| RTDS | input | - S plane denominator poly. root array (LCAP2 format) |
| PNW | output | - W plane numerator poly. coefficient array (LCAP2 format) |
| PDW | output | - W plane denominator poly. coefficient array (LCAP2 format) |
| RTNW | output | - W plane numerator poly. root array (LCAP2 format) |
| RTDW | output | - W plane denominator poly. root array (LCAP2 format) |
| T | input | - Sampling period |
| DELAY | input | - Time delay |
| IZOH | input | - .NE.0 for inclusion of zero order hold |

## Method

See description of SWXFM.

## Restrictions

See description of SWXFM.

## Requirements

COMMON blocks: HEADDB,INTCOM,PRNCTL,//
LCAP2 routines: CRELIM,PEQUAL,PMULT,PPADD,PPRINT,PPZERO,PROOT,PSYNTH,
RESDU,RPRINT,RREQU,RRTADD,RRTPRN,RRZERO,RTADD,RTPRNO,
RZERO,SUBP,TFRPRN

# WZXFM

## Identification

SUBROUTINE WZXFM - LCAP2 Operator, W to Z Plane Transformation
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Compute w to z plane bilinear transformation using an LCAP2 index. (note: the w is not the w' defined by the Tustin's bilinear rule)

## Usage

```
CALL WZXFM(I,J)
```

I input - Index of computed z plane transfer function
J input - Index of w plane transfer function to be transformed.

1. The sampling period, SAMPT, of COMMON/HEADDB/ must be set before calling this subroutine.

## Method

Bilinear transformation is implemented by transformation of the w plane roots. This method is more accurate than the method described by A.C. Davies (see description of BILNWZ).

## Requirements

COMMON blocks: INTCOM,PRNCTL,TFTEMP
LCAP2 routines: ENDLINE,FCNW1,FCNW2,FETTFX,HOLLI,OPMESG,PROOT,STFTFX,
TFPRN1,TFPRN4,WZTRANS,ZWTRANS

# XTRACT

## Identification

SUBROUTINE XTRACT - Extract Root Information From Packed Word
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Extract or unpack root information from the real part of a complex variable. The first word of a complex root array (LCAP2 format) contains this packed information.

## Usage

```
CALL XTRACT(ROOT,NUM,ICOM,IR,IZ)
```

| | | |
|---|---|---|
| ROOT | input | - Complex variable |
| NUM | output | - Total number of roots |
| ICOM | output | - Number of complex roots |
| IR | output | - Number of real roots not at the origin |
| IZ | output | - Number of roots at the origin |

## Method

The first word of a complex array, in LCAP2 format, is used to store information characterizing a polynomial. The real part of this word is packed as: (bit 1 is the unit digit, bit 2 is the tens digit, .. etc.)

| Decimal Digit | Description |
|---|---|
| 1,2 | Total number of roots |
| 3,4 | Number of complex roots (an even number) |
| 5,6 | Number of real roots not at the origin |
| 7,8 | Number of roots at the origin |

(Example: REAL(ROOT(1)) = 103048 would yield, NUM=8, ICOM=4, IR=3, IZ=1)

The imaginary part of the first word of a complex root array in LCAP2 format is the low order non-zero coefficient of the polynomial. This value is not affected by this subroutine.

## Requirements

COMMON blocks: none
LCAP2 routines: none

## ZELCR

### Identification

SUBROUTINE ZELCR - LCAP2 Operator, Eliminate Common Roots Of Z Plane
Transfer Function
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Eliminate common roots from a z plane transfer function using an LCAP2 index.

### Usage

```
CALL ZELCR(I)
```

I input - Index of z plane transfer function

1. Common root elimination parameters ECRE1 (preset=2.E-4) and ECRE2 (preset=1.E-8) are in COMMON/HEADDB/.

### Method

If a numerator root nrt and a denominator root drt are found such that ABS(drt/nrt - (1.,0.)).LT.ECRE1 for nrt.NE.0 or ABS(drt).LT.ECRE2 for nrt.EQ.0, roots nrt and drt are considered to be common roots and will be eliminated from the transfer function.

The code for this routine is in subroutine SELCR.

### Requirements

COMMON blocks: INTCOM,PRNCTL,TFTEMP
LCAP2 routines: CRELIM,ENDLINE,FCNPLN,FETTFX,OPMESG,OPTPRN,PROOT,PSYNTH,
RREQU,RTPRNO,STRTFX,TFPRN1,TFPRN4

## ZEQ1

### Identification

SUBROUTINE ZEQ1 - Find Number of Roots Equal to One
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Find the number of roots which are equal to or are nearly equal to (1.,0.). These roots are moved to the end of the list of non-zero roots.

### Usage

```
CALL ZEQ1(ROOT,L)
```

ROOT  input - Complex polynomial root array (LCAP2 format)
L  output - Number of roots = (1.,0.) found.

### Requirements

COMMON blocks: none
LCAP2 routines: XTRACT

# ZETAZXM

## Identification

SUBROUTINE ZETAZXM - Zeta To Z Transformation
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Compute transformation of zeta plane roots to z plane roots. This subroutine is used by subroutine MRXFM.

## Usage

```
CALL ZETAZXM(XRTF,YRTF,DELT)
```

| | | |
|---|---|---|
| XRTF | input | - Zeta plane polynomial root array (LCAP2 format) |
| YRTF | output | - Z plane polynomial root array (LCAP2 format) |
| DELT | input | - Reciprocal of radius defining zeta transform |

## Method

Zeta transformation is defined by:

$$z = (\text{zeta} * \text{radius}) + 1.$$

which maps the unit circle in the z plane into a circle in the left half zeta plane with the origin at zeta = -radius.

The code for this routine is in subroutine ZZETAXM.

## Requirements

COMMON blocks: none
LCAP2 routines: RCLAS,RTPRNO,XTRACT

# ZFAUX

## Identification

COMPLEX FUNCTION ZFAUX - Evaluate Z Plane Transfer Function Coefficient Array
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Evaluate z plane transfer function coefficient array (LCAP2 format) for use in computing the frequency response. This complex function can be used by subroutine FREQZ1 or FREQZ2 to evaluate the transfer function specified by its first argument. It can also be used by user-supplied subroutines similar to ZFAUX1.

This subroutine can also evaluate the multirate (fast input, slow output) response of the transfer function.

## Usage

```
ZFAUX(TFC)
```

TFC input - Transfer function coefficient array (LCAP2 format)
ZFAUX output - Complex value of response

1. Independent z plane frequency used in evaluation of the response is computed by the program using real frequency X of COMMON/FRQBLK/ and sampling period SAMPT of COMMON/HEADDB/.

2. If MMTGER of COMMON/FRQBLK/ is .GT.0, the multirate response is computed by using Sklansky's frequency decomposition method. MMTGER is the ratio of the (output/input) sampling periods and SAMPT is the sampling period of the faster input sampler.

## Requirements

COMMON blocks: FRQBLK,HEADDB,LENGTH
LCAP2 routines: none

# ZFAUX1

## Identification

COMPLEX FUNCTION ZFAUX1 - Evaluate Z Plane Transfer Function Coefficient Array
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

This complex function is similar to ZFAUX except that it is written so that it can be easily modified by the user to allow creation of a user-defined z plane transfer function.

## Usage

```
ZFAUX1(TFC)
```

TFC input - Transfer function coefficient array (LCAP2 format)
ZFAUX1 output - Complex value of the response

## Method

This complex function has only one line of code

```
ZFAUX1=ZFAUX(TFC)
```

so that it will yield the same results as WFAUX.

To create a user defined z plane transfer function, a different value is returned for ZFAUX1. For example, if the function is

```
ZPTF2 + ZPTF4/2.
```

the user would change the FORTRAN code to

```
ZFAUX1=ZFAUX(ZPTF2) + ZFAUX(ZPTF4)/2.
```

## Restrictions

The argument TFC must be an array in memory. Since only the first five z plane transfer functions are in COMMON/SCMBLK/ and all others are on a disk file, only ZPTF1, ZPTF2, ZPTF3, ZPTF4 and ZPTF5 can be used to create a user-defined z plane transfer function. However, the user can define additional transfer function coefficient arrays in a separate labeled common block to be accessible by ZFAUX. Subroutine FETZTF can be used to copy transfer functions from the disk file to the transfer function in this common block.

Requirements

COMMON blocks: FRQBLK,HEADDB,SCMBLK
LCAP2 routines: ZFAUX

# ZFREQ

## Identification

SUBROUTINE ZFREQ - LCAP2 Operator, Z Plane Frequency Response
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Evaluate z plane frequency response using an LCAP2 index. Automatic frequency mode available to allow program to dynamically choose its own frequency points to yield a smooth plot of the response.

## Usage

```
CALL ZFREQ(I)
```

I input - Index of z plane transfer function

1. Frequency response parameters are in COMMON/HEADDB. They are to be set before ZFREQ is called. See description of SFREQ for the complete list of definitions of these parameters. The parameter SAMPT is described below:

| parameter | preset | description |
|---|---|---|
| SAMPT | 1 | Sampling period |

## Method

Same as that described in detail in description of SFREQ.

The code for this routine is in subroutine SFREQ.

## Requirements

COMMON blocks: FRQBLK,INTCOM,PRNCTL,TFTEMP,//
LCAP2 routines: ENDLINE,FCNPLN,FETTFX,FREQS2,FREQW2,FREQZ2,OPMESG,SFAUX,
WFAUX,ZFAUX

# ZFREQY

## Identification

SUBROUTINE ZFREQY - Evaluate Frequency Response Of A Z Plane Transfer Function Coefficient Array
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Evaluate frequency response of a z plane transfer function coefficient array. User supplies name of the array.

## Usage

```
CALL ZFREQY(TFC)
```

TFC input - Transfer function coefficient array (LCAP2 format)

1. Frequency response parameters are in COMMON/HEADDB/. See description of subroutine SFREQ for definition.

## Restrictions

If LCAP2 defined transfer function coefficient arrays are to be used, only the first five transfer functions for each plane are available, since the others are on disk files. However, a user common block can be defined so that these other transfer functions can be first transferred from disk file to memory so that ZFREQY can be used.

## Requirements

COMMON blocks: none
LCAP2 routines: FREQZ2,ZFAUX

# ZHOLLI

## Identification

FUNCTION ZHOLLI - Converts Integer To Hollerith Characters, Zero Filled On The Left
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Converts integer to Hollerith characters, zero filled on the left.

## Usage

```
ZHOLLI(I)

I       input  - Integer between 0 and 100, or else -1 or -2
ZHOLLI  output - Hollerith representation of I if I is between 0 and 100,
                 (zero filled on the left if I is one digit)
                 = 2HN  if I=-1
                 = 2HD  if I=-2
                 = Blank otherwise
```

## Method

The code for this function is in function HOLLI.

## Requirements

COMMON blocks: none
LCAP2 routines: none

## ZLOCI

### Identification

SUBROUTINE ZLOCI - LCAP2 Operator, Z Plane Root Locus
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Evaluate z plane root locus using an LCAP2 index. Automatic gain selection available to supplement user-selected gains.

### Usage

```
CALL ZLOCI(I)
```

I input - Index of z plane transfer function to be evaluated

1. Root locus parameters are in COMMON/HEADDB/. They are to be set before ZLOCI is called. See description of SLOCI for a complete list of definitions of these parameters.

### Method

Same as that described in detail in description of SLOCI.

The code for this routine is in subroutine SLOCI.

### Requirements

COMMON blocks: INTCOM,PRNCTL,TFTEMP,//
LCAP2 routines: ENDLINE,FCNPLN,FETTFX,OPMESG,RLOCUS1,TFPRN1

# ZMFAUX

## Identification

COMPLEX FUNCTION ZMFAUX - Evaluate Multirate Z Plane Transfer Function
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Evaluate multirate (fast input, slow output) response of a z plane transfer function coefficient array for use in computing the frequency response. This complex function can be used by subroutines FREQZM1 or FREQZM2 to evaluate the transfer function specified by its first argument.

## Usage

ZMFAUX(TFC,M,T)

| | | |
|---|---|---|
| TFC | input | - Transfer function coefficient array (LCAP2 format) |
| M | input | - Integer ratio of output/input sampling periods |
| T | input | - Sampling period of slower output sampler |
| ZMFAUX1 | output | - Complex value of response |

## Method

Sklansky's frequency decomposition method is used to evaluate the transfer function.

## Requirements

COMMON blocks: FRQBLK,HEADDB
LCAP2 routines: ZFAUX

# ZMRFQ

## Identification

SUBROUTINE ZMRFQ - LCAP2 Operator, Z Plane Multirate Frequency Response
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Evaluate multirate (fast input, slow output) frequency response of a z plane transfer function using an LCAP2 index.

## Usage

```
CALL ZMRFQ(I,M)
```

I input - Index of z plane transfer function
M input - Integer ratio of output/input sampling periods

1. The input z plane transfer function is at the faster sampling rate.

2. Frequency response parameters are in COMMON/HEADDB/. See description of SFREQ.

3. The sampling period, SAMPT, is for the slower output sampler.

## Method

The frequency response is evaluated by direct application of Sklansky's frequency decomposition. No explicit rational representation of the slower output transform is computed. If an explicit representation of the slower output transfer function is desired, see LCAP2 operator ZMRXFM.

The code for this routine is in subroutine WMFRQ.

## Requirements

COMMON blocks: INTCOM,FRQBLK,PRNCTL,TFTEMP
LCAP2 routines: ENDLINE,FCNPLN,FCNW1,FETTFX,FREQWM2,FREQZM2,OPMESG,
TFPRN1,WFAUX,ZFAUX

# ZMRFRQY

## Identification

SUBROUTINE ZMRFQY - Evaluate Multirate Frequency Response Of A Z Plane
Transfer Function Coefficient Array
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Evaluate multirate (fast input, slow output) frequency response of a z plane transfer function coefficient array. User supplies the name of the array.

## Usage

```
CALL ZMRFQY(TFC,M)
```

TFC input - Transfer function coefficient array (LCAP2 format) at the faster input sampling rate
M input - Integer ratio of output/input sampling periods

1. Frequency response parameters are in COMMON/HEADDB/. See description of SFREQ.

2. The sampling period, SAMPT, is for the slower output sampler.

## Requirements

COMMON blocks: FRQBLK
LCAP2 routines: FREQZM2,ZFAUX

## ZMRXFM

### Identification

SUBROUTINE ZMRXFM - LCAP2 Operator, Multirate (fast input, slow output sampler) Z Transform
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Compute the output z transform of a fast to slow sampler using LCAP2 indices. This operation will yield a rational transfer function at the slower sampling rate.

### Usage

```
CALL ZMRXFM(I,J)
```

I input - Index of resultant slower output z transfer function
J input - Index of faster input z transfer function

1. The integer ratio, output/input sampling periods, NTGER, of COMMON/HEADDB/ must be set before this subroutine is called.

### Method

The output transform of a fast to slow rate sampler is given by Sklansky's frequency decomposition method as

$$\frac{1}{n}\sum_{k=1}^{n-1} G_n\left(z^{T/n} e^{j2*pi*k/n}\right)$$

where $G_n(z^{T/n})$ is the z plane transfer function at the faster sampling rate and n is the integer ratio of the output/input sampling periods. Using the root form representation of the input transform, a rational representation of the slower rate output transform is computed.

### Requirements

COMMON blocks: HEADDB,ITEST,PRNCTL,TFTEMP
LCAP2 routines: CRELIM,ENDLINE,FETTFX,MRXFM,OPPRN,PROOT,PSYNTH,RREQU,
RRTEQU,RTPRNO,STRTFX,TFPRN1,TFPRN4,WZTRANS,ZWTRANS

# ZNORM

## Identification

SUBROUTINE ZNORM - LCAP2 Operator, Normalize Z Plane Transfer Function
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Normalize z plane transfer function using an LCAP2 index. Normalization can be either with respect to the low order non-zero coefficient or the high order coefficient of the denominator.

## Usage

```
CALL ZNORM(I)
```

I input - Index of the z plane transfer function

1. Normalization parameters are in COMMON/HEADDB/. They are to be set before ZNORM is called. These parameters are defined below:

| parameter | preset | description |
|---|---|---|
| KNORM | 1. | Value used for normalizing the transfer function |
| KNRMFG | 0 | If .EQ.0, the low order non-zero coefficient of the denominator is set equal to the value of KNORM and all other coefficients are normalized to this value.<br><br>If .NE.0, the high order coefficient of the denominator is set equal to the value of KNORM and all other coefficients are normalized to this value. |

## Method

The code for this routine is in subroutine SNORM.

## Restrictions

KNORM cannot be zero.

## Requirements

COMMON blocks: INTCOM,PRNCTL,TFTEMP
LCAP2 routines: ENDLINE,FCNPLN,FETTFX,NORM,OPMESG,STRTFX,TFPRN1,TFPRN4

# ZPADD

## Identification

SUBROUTINE ZPADD - LCAP2 Operator, Z Plane Transfer Function Add
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Add two z plane transfer functions using LCAP2 indices.

## Usage

```
CALL ZPADD(I,J,K)
```

I input - Index of resultant transfer function sum
J input - Index of first transfer function to be added
K input - Index of second transfer function to be added

## Method

The code for this routine is in subroutine SPADD.

## Restrictions

The degree of the transfer function must be less than 50.

## Requirements

COMMON blocks: PRNCTL,TFTEMP
LCAP2 routines: ENDLINE,FCNPLN,FCNW1,FCNW2,FETTFX,OPPRN,PPADD,RRTADD,
STRTFX,TFPRN1,TFPRN4

# ZPDIV

## Identification

SUBROUTINE ZPDIV - LCAP2 Operator, Z Plane Transfer Function Divide
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Divide two z plane transfer functions using LCAP2 indices.

## Usage

```
CALL ZPDIV(I,J,K)
```

I input - Index of resultant transfer function
J input - Index of dividend transfer function
K input - Index of divisor transfer function

## Method

The code for this routine is in subroutine SPDIV.

## Restrictions

The degree of the transfer functions must be less than 50.

## Requirements

COMMON blocks: PRNCTL,TFTEMP
LCAP2 routines: ENDLINE,FETTFX,OPPRN,PMULT,PSYNTH,RTMPY,STRTFX,
TFPRN1,TFPRN4

# ZPEQU

## Identification

SUBROUTINE ZPEQU - LCAP2 Operator, Z Plane Equal
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Equate z plane transfer functions using LCAP2 indices.

## Usage

CALL ZPEQU(I,J)

I input - Index of resultant transfer function
J input - Index of transfer function to be equated with

## Method

The code for this routine is in subroutine SPEQU.

## Restrictions

The degree of the transfer functions must be less than 50.

## Requirements

COMMON blocks: OVCOM,PRNCTL,TFTEMP
LCAP2 routines: ENDLINE,FETTFX,OPPRN,PPEQU,RRTEQU,STFTFX,TFPRN1,TFRPRN4

# ZPLDC

## Identification

SUBROUTINE ZPLDC - LCAP2 Operator, Load Coefficients Into Z Plane Transfer Function

CDC FORTRAN 4

E. A. Lee
Aerospace Corporation

## Purpose

Load coefficients into z plane transfer function using an LCAP2 index.

## Usage

```
CALL ZPLDC(I)
```

I input - Index where transfer function is to be stored

1. Transfer function coefficients are entered with polynomial coefficient arrays POLYN and POLYD (LCAP2 format) which are in COMMON/HEADDB/. They are to be set before ZPLDC is called.

2. The calling program must include COMMON/HEADDB/ and the appropriate DIMENSION and EQUIVALENCE statements for POLYN and POLYD.

3. The roots of ZPTFi will not be automatically computed. If this is desired, follow this operation with the operator ZPRTS(I).

## Method

The code for this routine is in subroutine SPLDC.

## Restrictions

The degree of the transfer function must be less than 50.

## Requirements

COMMON blocks: INTCOM,HEADDB,PRNCTL,TFTEMP
LCAP2 routines: ENDLINE,FCNPLN,OPMESG,PEQUAL,PPRN1,STRTFX,TFPRN4

## ZPLDR

### Identification

SUBROUTINE ZPLDR - LCAP2 Operator, Load Z Plane Transfer Function In Root Form
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Load roots into z plane transfer function using an LCAP2 index.

### Usage

```
CALL ZPLDR(I)
```

I input - Index where transfer function is to be stored

1. Transfer function roots are entered with polynomial coefficient arrays ROOTN and ROOTD (LCAP2 format) which are in COMMON/HEADDB/. They are to be set before ZPLDR is called.

2. The calling program must include COMMON/HEADDB/ and the appropriate DIMENSION and EQUIVALENCE statements for ROOTN and ROOTD.

### Method

The code for this routine is in subroutine SPLDR.

### Restrictions

The degree of the transfer function must be less than 50.

### Requirements

COMMON blocks: INTCOM,HEADDB,PRNCTL,TFTEMP
LCAP2 routines: ENDLINE,FCNPLN,OPMESG,PSYNTH,RTEQU,RTPRN2,STRTFX,TFPRN4

# ZPMPY

## Identification

SUBROUTINE ZPMPY - LCAP2 Operator, Z Plane Transfer Function Multiply
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Multiply two z plane transfer functions using LCAP2 indices.

## Usage

```
CALL ZPMPY(I,J,K)
```

I input - Index of resultant transfer function product
J input - Index of first transfer function to be multiplied
K input - Index of second transfer function to be multiplied

## Method

If only the coefficients of the j-th and k-th transfer functions are available, the product is computed by multiplication of the coefficients. If the roots of the j-th and k-th transfer functions are available, the product is computed by combining the roots. The coefficients of the product are then formed from these roots.

The code for this routine is in subroutine SPMPY.

## Restrictions

The degree of the transfer functions must be less that 50.

## Requirements

COMMON blocks: PRNCTL,TFTEMP
LCAP2 routines: ENDLINE,FCNW1,FCNW2,FETTFX,OPPRN,PMULT,PSYNTH,RTMPY,
STRTFX,TFPRN1,TFPRN4

## ZPPRN

### Identification

SUBROUTINE ZPPRN - LCAP2 Operator, Print Out Z Plane Transfer Function
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Print out z plane transfer function using an LCAP2 index.

### Usage

```
CALL ZPPRN(I)
```

I input - Index of transfer unction to be printed out

### Method

Roots of the transfer function are printed out only if they are defined (previously computed or loaded in). The coefficients of the transfer function are printed out in ascending order.

The code for this routine is in subroutine SPPRN.

### Requirements

COMMON blocks: INTCOM,TFTEMP
LCAP2 routines: ENDLINE,FCNPLN,FETTFX,OPMESG,TFPRN4

# ZPRTS

## Identification

SUBROUTINE ZPRTS - LCAP2 Operator, Find Roots Of Z Plane Transfer Function
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Find roots of a z plane transfer function using an LCAP2 index.

## Usage

```
CALL ZPRTS(I)
```

I input - Index of z plane transfer function

## Method

Roots of the numerator and denominator are computed by subroutine PROOT.

The code for this routine is in subroutine SPRTS.

## Restrictions

If the roots of ZPTFi were previously computed or loaded in, the program will not recompute the roots from the coefficients. A message to this effect will be printed.

## Requirements

COMMON blocks: INTCOM,PRNCTL,TFTEMP
LCAP2 routines: ENDLINE,FCNPLN,FCNW1,FETTFX,OPMESG,PROOT,STRTFX,TFPRN1,
TFPRN4

# ZPSUB

## Identification

SUBROUTINE ZPSUB - LCAP2 Operator, Z Plane Transfer Function Subtract
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Subtract two z plane transfer functions using LCAP2 indices.

## Usage

```
CALL ZPSUB(I,J,K)
```

I input - Index of resultant transfer function difference
J input - Index of first transfer function (minuend)
K input - Index of second transfer function (subtrahend)

## Method

The code for this routine is in subroutine SPSUB.

## Restrictions

The degree of the transfer functions must be less than 50.

## Requirements

COMMON blocks: PRNCTL,TFTEMP
LCAP2 routines: ENDLINE,FETTFX,OPPRN,PPSUB,RRTSUB,STRTFX,TFPRN1,TFPRN4

# ZSXFM

## Identification

SUBROUTINE ZSXFM - LCAP2 Operator, Transform Z Plane Roots Into S Plane
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Transform z plane roots into "equivalent" s plane roots using an LCAP2 index. The transformation of the z plane roots to the s plane is not unique. The "equivalent" s plane roots are provided solely to aid the analyst in identifying and correlating z plane roots. The computed s plane roots are not saved.

## Usage

```
CALL ZSXFM(I)
```

I input - Index of z plane transfer function

1. Sampling period, SAMPT, of COMMON/HEADDB/ must be set before calling this subroutine.

## Method

Transformation of the roots from z to the s plane is defined by

$$s = \ln( z ) / SAMPT$$

The code for this routine is in subroutine WSXFM.

## Requirements

COMMON blocks: INTCOM,HEADDB,PRNCTL,TFTEMP
LCAP2 routines: ENDLINE,FCNPLN,FCNW1,FCNW2,FETTFX,OPMESG,TFPRN1,WSTRAN1

# ZTIME

## Identification

SUBROUTINE ZTIME - LCAP2 Operator, Inverse Z Transform And Time Response
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Compute inverse z transform and the time response using an LCAP2 index.

## Usage

```
CALL ZTIME(I)
```

I input - Index of z plane transfer function

1. Time response parameters are in COMMON/HEADDB/. They are to be set before ZTIME is called. These parameters are defined below:

| parameter | preset | description |
|---|---|---|
| TSTEP | 1 | .NE.0 for step response; .EQ.0 for impulse response |
| TMAGN | 1. | Magnitude of input for time response |
| TEND | 1. | End time for evaluating time response |
| TMAGN | 1. | Magnitude of input for time response |
| SAMPT | 1. | Sampling period |
| GRAFP | 1 | .NE.0 for printer (low resolution) plot |
| FILM | 0 | .NE.0 for hardcopy (high resolution) plot |
| TXMIN | 0 | Minimum x axis for plot |
| TXMAX | 0 | Maximum x axis for plot (Auto. scaling of x axis if TXMIN=TXMAX) |
| TYMIN | 0 | Minimum y axis for plot |
| TYMAX | 0 | Mamimum y axis for plot (Auto. scaling of y axis if TYMIN=TYMAX) |

## Method

The inverse z transform is computed by the power series (long division) method. While this method of computing the time response is inherently less accurate than the partial fraction method, results for typical transfer functions are excellent. To provide a measure of the accuracy of the response, the results are computed in double precision and compared.

The code for this routine is in subroutine STIME.

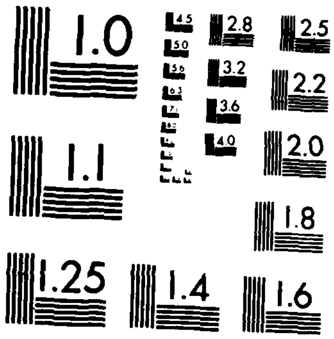
## Restrictions

The degree of the transfer function must be less than 50.

**Requirements**

COMMON blocks: INTCOM,PRNCTL,TFTEMP,//
LCAP2 routines: ENDLINE,FCNPLN,FCNW1,FCNW2,FETTFX,OPMESG,PROOT,STIME1,
STRTFX,TFPRN1,ZTIME1

AD-A141 386 LCAP2 (LINEAR CONTROLS ANALYSIS PROGRAM) VOLUME 3 4/4
SOURCE CODE DESCRIPTION..(U) AEROSPACE CORP EL SEGUNDO
CA GUIDANCE AND CONTROL DIV E A LEE 15 NOV 83
UNCLASSIFIED TR-0084(9975)-1-VOL-3 SD-TR-84-06-VOL-3 F/G 9/2 NL


1.0
4.5
5.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

# ZTIME1

## Identification

SUBROUTINE ZTIME1 - Inverse Z Transform and Time Response
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Compute inverse z transform and time response. This subroutine is called by ZTIME.

## Usage

```
CALL ZTIME1(PNS,PDS,RTNS,RTDS)
```

| | | |
|---|---|---|
| PNS | input | - Numerator polynomial coefficient array (LCAP2 format) |
| PDS | input | - Denominator polynomial coefficient array (LCAP2 format) |
| RTNS | input | - Complex numerator polynomial root array (LCAP2 format) |
| RTDS | input | - Complex denominator polynomial root array (LCAP2 format) |

1. Time response parameters are in COMMON/HEADDB/. See description of ZTIME.

## Method

Subroutine ZTIME2 is called to evaluate the time response.

For interactive LCAP2 the user is prompted for the type of input and the beginning and end times to be used for evaluating the response. The user is also given the option to suppress tabular output of the response.

The code for this routine is in subroutine STIME1.

## Requirements

COMMON blocks: CMELIM,CMRESD,CMTIME,HEADDB,INTCOM,//
LCAP2 routines: DAYPRN,GRAF1,ITITLE,STIME2,STIME3,TSPLOT,ZTIME2

# ZTIME2

## Identification

SUBROUTINE ZTIME2 - Inverse Z Transform By Power Series Method
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Compute the inverse z transform by the power series method.

## Usage

```
CALL ZTIME2(PNZ,PDZ,RTNZ,RTDZ,TPOINT,YPOINT,ZPOINT,NTERM)
```

| | | |
|---|---|---|
| PNZ | input | - Numerator polynomial coefficient array (LCAP2 format) |
| PDZ | input | - Denominator polynomial coefficient array (LCAP2 format) |
| RTNZ | | - not used |
| RTDZ | | - not used |
| TPOINT | output | - Array of time points (DIMENSION=1500) |
| YPOINT | output | - Array of time response points (DIMENSION=1500) |
| ZPOINT | output | - Array of time response points (DIMENSION=1500) |
| NTERM | output | - Number of time points computed (max=1500) |

1. Time response parameters are in COMMON/HEADDB/. See description for STIME.

## Method

The inverse z transform is computed by the power series (long division) method. While this method of computing the time response is inherently less accurate than the partial fraction method, results for typical transfer functions are excellent. To provide a measure of the accuracy of the response, the results are computed in double precision and compared.

## Restrictions

Degree of the numerator must not be greater than the denominator.

## Requirements

COMMON blocks: HEADDB,INTCOM
LCAP2 routines: none

## ZVCHNG1

### Identification

SUBROUTINE ZVCHNG1 - Z to (Z**n) Transformation
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Compute transformation of the z plane transfer function to a faster z variable by replacement of variables. Called by ZVCNG.

### Usage

```
CALL ZVCHNG1(TFPZI,TFRZI,INI,IDI,TFPZJ,TFRZJ,INJ,IDJ)
```

| | | |
|---|---|---|
| TFPZI | input | - Transfer function coefficient array (LCAP2 format) at the slower sampling rate |
| TPRZI | input | - Transfer function root array (LCAP2 format) at the slower sampling rate |
| INI | input | - =1 for numerator coefficient form only<br>=0 for numerator coefficient and root form |
| IDI | input | - =1 for denominator coefficient form only<br>=0 for denominator coefficient and root form |
| TFPZJ | output | - Transfer function coefficient array at the faster sampling rate |
| TPRZJ | output | - Transfer function root array at the faster sampling rate |
| INJ | output | - =1 for numerator coefficient form only<br>=0 for numerator coefficient and root form |
| IDJ | output | - =1 for denominator coefficient form only<br>=0 for denominator coefficient and root form |

### Method

See description for ZVCNG

### Restrictions

See description of ZVCNG.

### Requirements

COMMON blocks: HEADDB
LCAP2 routines: LEXIT,PPZERO

## ZVCNG

### Identification

SUBROUTINE ZVCNG - LCAP2 Operator, Z to (Z**n) Transformation
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Compute transformation of z plane transfer function to a faster z variable by replacement of variables.

### Usage

```
CALL ZVCNG(I,J)
```

I input - Index of resultant z plane transfer function expressed in terms of the faster z variable
J input - Index of z plane transfer function to be operated upon

1. The integer ratio of faster/slower sampling rate, NTGER, of COMMON/HEADDB/ must be set before ZVCNG is called.

### Method

The z variable of the slower sampled transfer function is replaced by z**n and stored into the faster sampled transfer function. This subroutine calls ZVCHNG1.

### Restrictions

Since the format used to represent transfer function arrays in LCAP2 limits the degree of the polynomials to less than 50, the degree of the j-th z plane transfer function times NTGER must be less than 50.

### Requirements

COMMON blocks: PRNCTL,TFTEMP
LCAP2 routines: DOTLINE,ENDLINE,FCNW1,FETTFX,HOLLI,STRTFX,TFPRN1,ZVCHNG1

# ZWTRANS

## Identification

SUBROUTINE ZWTRANS - Bilinear Transformation of Z Plane Roots to W Plane Roots
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Compute bilinear transformation of z plane roots to w plane roots.

## Usage

```
CALL ZWTRANS(AROOT,ATF,BROOT,BTF)
```

AROOT input - Z plane transfer function root array (LCAP2 format)
ATF input - Z plane transfer function coefficient array (LCAP2 format)
BROOT output - W plane transfer function root array (LCAP2 format)
BTF output - W plane transfer function coefficient array (LCAP2 format)

## Method

Roots are transformed from the z plane to the w plane by the following:

$$w = (z-1) / (z+1)$$

The low order non-zero coefficients of the numerator and denominator of the z plane transfer function are computed so that the correct gain is maintained when the transfer function is evaluated.

## Requirements

COMMON blocks: none
LCAP2 routines: PPZERO,PSYNTH,RCLAS,RRZERO,RTEQ1,RTEQ2,XTRACT

# ZWXFM

## Identification

SUBROUTINE ZWXFM - LCAP2 Operator, Z to W Plane Transformation
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

## Purpose

Compute z to w plane bilinear transformation using an LCAP2 index. (note: the w is not the w' defined by the Tustin's bilinear rule)

## Usage

```
CALL ZWXFM(I,J)
```

I input - Index of computed w plane transfer function
J input - Index of z plane transfer function to be transformed.

1. The sampling period, SAMPT, of COMMON/HEADDB/ must be set before calling this subroutine.

## Method

Bilinear transformation is implemented by transformation of the z plane roots. This method is more accurate than the method described by A.C. Davies (see description of BILNWZ).

The code for this routine is in subroutine WZXFM.

## Requirements

COMMON blocks: INTCOM,PRNCTL,TFTEMP
LCAP2 routines: ENDLINE,FCNW1,FCNW2,FETTFX,HOLLI,OPMESG,PROOT,STRTFX,
TFPRN1,TFPRN4,WZTRANS,ZWTRANS

## ZZETAXM

### Identification

SUBROUTINE ZZETAXM - Z to Zeta Transformation
CDC FORTRAN 4
E. A. Lee
Aerospace Corporation

### Purpose

Compute transformation of z plane roots to zeta plane roots. This subroutine is used by subroutine MRXFM.

### Usage

```
CALL ZZETAXM(XRTF,YRTF,DELT)
```

XRTF input - Z plane polynomial root array (LCAP2 format)
YRTF output - Zeta plane polynomial root array (LCAP2 format)
DELT input - Reciprocal of radius defining zeta transform

### Method

Zeta transformation is defined by:

$$z = (\text{zeta} * \text{radius}) + 1.$$

which maps the unit circle in the z plane into a circle in the left half zeta plane with the origin at zeta = -radius.

### Requirements

COMMON blocks: none
LCAP2 routines: RCLAS,RTPRNO,XTRACT

REFERENCES

1. E. A. Lee, "Linear Controls Analysis Program (LCAP) Users Guide," Aerospace Corporation, TOR - 0077(2442-23)-1, 5 October 1976.

2. E. A. Lee, "LCAP2 - Linear Control Analysis Program, Vol I: Batch LCAP2 Users Guide," Aerospace Corporation, TR - 0084(9975)-1 Vol I, 15 November 1983.

3. E. A. Lee, "LCAP2 - Linear Control Analysis Program, Vol II: Interactive LCAP2 Users Guide," Aerospace Corporation, TR - 0084(9975)-1 Vol II, 15 November 1983.

4. J. F. Holt, "ASC MULE General Root Finding Subroutine," Aerospace Corporation, TOR - 0073(9320)-8, 23 March 1973.

# APPENDIX A. COMDECK LCAP2 FOR PROGRAM UPDATE

The FORTRAN code defining COMDECK LCAP2 for program UPDATE used in creating the main program for batch LCAP2 is given below:

```
      PROGRAM LCAP2(INPUT,O[illegible]PUT=/120,TAPE5=INPUT,TAPE6=OUTPUT
     +,TAPE19,TAPE30,TAPE31,[illegible]APE83,TAPE84,TAPE85,TAPE86
     +,TAPE87,TAPE89
     +)
      COMMON/FRQBLK/U,X,TWOPI,MMTGER
      COMPLEX X
      COMMON/LENGTH/NDEGP1,NDEGP2
      COMMON/PLOT1/NPLOTS
      COMMON/PRNCTL/PRNFLG1,PRNFLG2,PRNFLG3,PRNFLG4,PRNFLG5
      COMMON/TFPCNT/NSPCNT,NWPCNT,NZPCNT,NPYCNT
      COMMON/TFTEMP/TFPI(102),TFRI(100),INI,IDI,TFPJ(102),TFRJ(100),INJ
     +,IDJ,TFPK(102),TFRK(100),INK,IDK
      COMPLEX TFRI,TFRJ,TFRK
      COMMON/HEADDB/HEAD(70),DB(900)
      DIMENSION POLYN(51),POLYD(51),POLY(51)
      COMPLEX OMEGA(20),FREQ1(3),FREQ2(3),FREQ3(3),FREQ4(3),FREQ5(3)
      COMPLEX ROOTN(50),ROOTD(50),ROOT(50)
      EQUIVALENCE (DB(296),POLYN),(DB(347),POLYN),(DB(498),ROOTN)
     +,(DB(598),ROOTD),(DB(245),POLY),(DB(398),ROOT)
      EQUIVALENCE (DB(101),CONTP),(DB(103),CYCLE),(DB(104),DBMAX)
     +,(DB(105),DBMIN),(DB(106),DEGMN),(DB(107),DELAY),(DB(108),ECRE1)
     +,(DB(109),ECRE2),(DB(110),EDB1 ),(DB(111),EDB2 ),(DB(112),EDEG1)
     +,(DB(113),EDEG2),(DB(114),EPAD1),(DB(115),EPMR1),(DB(116),FAUX )
     +,(DB(117),EP1  ),(DB(118 ,EP2  ),(DB(119 ,EP3  ),(DB(120),EP4  )
     +,(DB(121),EP5  ),(DB(122),ERCNJ),(DB(123),ERCX ),(DB(124),ERCZ )
     +,(DB(126),FDLAY),(DB(127),FXYDL),(DB(128),FXYMN),(DB(129),FAUTO)
     +,(DB(130),FBODE),(DB(131),FILM ),(DB(132),FNICO),(DB(133),FNYQS)
     +,(DB(134),GRAFP),(DB(135),ITLOC),(DB(136),NTGER),(DB(140),KDELT)
     +,(DB(141),KFLG ),(DB(142),MAXIT),(DB(143),MNDW ),(DB(144),MXDW )
     +,(DB(145),MXITF),(DB(146),NLOCI),(DB(147),NOMEG),(DB(148),NP   )
     +,(DB(149),PMARG),(DB(151),PRN1 ),(DB(152),PRN2 ),(DB(153),PRN3 )
     +,(DB(154),PRN4 ),(DB(155),PRN5 ),(DB(156),PRN6 )
      EQUIVALENCE (DB(157),PRN7),(DB(158),PRN8),(DB(159),PRN9)
     +,(DB(160),PRN10),(DB(161),PSTOP),(DB(162),PTYPE),(DB(163),SAMPT)
     +,(DB(164),SHADE),(DB(165),TDELT),(DB(166),TEND ),(DB(167),TMAGN)
     +,(DB(169),TSTEP),(DB(170),TXMAX),(DB(171),TXMIN),(DB(172),TYMAX)
     +,(DB(173),TYMIN),(DB(174),TXTRA),(DB(175),TZERO),(DB(176),NANOT)
     +,(DB(177),XGAP ),(DB(178 ,YANOT),(DB(179 ,YGAP ),(DB(180),ZLINE)
     +,(DB(181),ZOH  ),(DB(182),NRMFG),(DB(183),KNORM),(DB(184),KCLP )
     +,(DB(185),ARFLG),(DB(186),RTMAX),(DB(187),RZERO),(DB(188),RLYMN)
     +,(DB(189),RLXMX),(DB(190),RLYMN),(DB(191),RLYMX),(DB(192),RLFG1)
     +,(DB(193),TZFLG),(DB(194),TZEP1),(DB(195),RAD  ),(DB(199),NTGER)
     +,(DB(137),XNCOL),(DB(138),XLINES),(DB(102),NQDB)
     +,(DB(200),KGAIN),(DB(225),OMEGA),(DB(698),FREQ1),(DB(701),FREQ2)
     +,(DB(704),FREQ3),(DB(707),FREQ4),(DB(710),FREQ5)
```

```
      REAL ITLOC,MTGER,KDELT,KFLG,MAXIT,MNDW.MXDW,MXITF,NLOCI,NOMEG
     +,NP,NRMFG,KNORM,KCLP,NTGER,NQDB,KGAIN(25)
      COMMON/INTCOM/INTFLG,NSTORE,NDETRM
      COMMON/MATRIX1/MATDIM,MXM,MDEG,MGESS(50),M0(30,30),M1(30,30)
     +,M2(30,30),M3(30,30),M4(30,30)
      REAL MXM,MDEG,M0,M1,M2,M3,M4
      COMPLEX MGESS
      COMPLEX DET,AMATRIX
      COMMON/MDET1/NR,DET,NDIMA,AMATRIX(30,30)
C
      INTEGER OVARG,OVENTRY
      COMMON/OVCOM/OVARG,OVENTRY,IOV,JOV,KOV,IFAUX
     +,OVCNTR1,OVCNTR2,OVCNTR3,OVCNTR4,OVCNTR5
      COMMON/SCMBLK/XTFS(1520),XTFW(1520),XTFZ(1520),XPY(760)
      COMMON/CMPOLY/POLYC(51)
```

## APPENDIX B. SEGLOAD DIRECTIVES FOR LOADING INTERACTIVE LCAP2 ROUTINES

The directives for defining the tree structure for interactive LCAP2 is given below:

```
        TREE LCAP2-ILCAP2-(IARG0,IARG1A,IARG2,INTXFM,IARG1B)
*       SEGLOADER WILL NOT WORK PROPERLY IF SUBROUTINES WHICH ARE
*       PASSED AS ARGUMENTS IN CALLING STATEMENTS ARE INCLUDED IN
*       THE SEGLOADER DIRECTIVES. EXAMPLES ARE SFAUX,ZFAUX,FAUXW,
*       ETC. SEGLOADER WILL PUT THESE SUBROUTINES IN THEIR CORRECT
*       PLACES.
LCAP2   INCLUDE LEVEL1
LCAP2   INCLUDE PLOTS
LCAP2   INCLUDE STDGRD
LCAP2   INCLUDE PLTSYM
LCAP2   INCLUDE SYMBOL
LCAP2   INCLUDE NUPLOT
LCAP2   INCLUDE BUFF
*
        GLOBAL FRQBLK,ACOM,LENGTH,PLOT1,PRNCTL,TFPCNT,TFTEMP
        GLOBAL HEADDB,OVCOM,SCMBLK,CMPOLY,AWORDS,INTCOM
        GLOBAL CMELIM,CMRESD,CMTIME,TEMPRT,ITEST,COMAXX
        GLOBAL SEQCOM,CALCOM,PINOUC,PLBUFF
*
        LEVEL
*
        TREE SPLDC
        TREE SFREQ
        TREE SFSQR
        TREE WSXFM
        TREE SWMRX
        TREE SWXFM
        TREE SZXFM
        TREE ZVCNG
        TREE WZXFM
        TREE SPLDR
        TREE SPADD
        TREE SPEQU
        TREE SPSUB
        TREE SPMPY
        TREE SPDIV
        TREE SNORM
        TREE SELCR
        TREE SPPRN
        TREE SPRTS
        TREE STIME
        TREE SLOCI
        TREE WMRFQ
        TREE PLDC
        TREE PLDR
```

```
         TREE PPRN
         TREE PRTS
         TREE PADD
         TREE PEQU
         TREE PMPY
         TREE CPYPS
         TREE CPYSP
         TREE PSUB
         TREE ZFREQY
         TREE ZMRXFM
*        ***************************************************************
         LEVEL
*
         TREE COEFF
         TREE COEFP
         TREE DOTLINE
         TREE ENDLINE
         TREE FCNPLN
         TREE FREQS-FREQS2-FPLOT1
         TREE IDETRM
         TREE IDTERM
         TREE INITO
         TREE MRXFM
         TREE NORM
         TREE OPMESG
         TREE IRSTOR
         TREE IROOT
         TREE IROOTP
         TREE RLOCUS1
         TREE STIME1-TSPLOT
         TREE ISTORE
         TREE WSTRAN1
         TREE WTRANS
         TREE ZWTRANS
         TREE ZWTRAN
         TREE ZTIMEX
         TREE ZVCHNG1
*        ************************************************
         LEVEL
*
*        DO NOT INCLUDE AXXMRN IN SEGMENT LOADER
         TREE EJK
         TREE BILNZW
         TREE EVMRRT
         TREE FETPY
         TREE FETTFX-FETSTF
         TREE ELPLOT1
         TREE MATROT2
         TREE FREQS3
         TREE MINIT2
```

```
         TREE MROOT2-CXMTX1
*        DO NOT INCLUDE AUXM2 IN SEGMENT LOADER
*        LET SEGMENT LOADER DO IT
         TREE OPPRN
         TREE RLOCIN1
         TREE RRTADD
         TREE RTEQ1
         TREE TFPRN1-TFPRN4-PYPRN1-PYPRN4-RTPRN1-RTPRN2
         TREE STIME2-STIME3-RESDU
         TREE STRTFX-STRSTF
         TREE PPADD
         TREE ZTIME2
         TREE ZZETAXM
*        *********************************************************
         LEVEL
*
         TREE ADDP
         TREE BPRINT2
         TREE CRELIM
         TREE EVLRT
         TREE FREQSI1
         TREE MPRINT2
         TREE PROOT
         TREE PSYNTH
         TREE PMULT
         TREE PPRINT
         TREE DAYPRN
         TREE FCNW1
         TREE RTADD
         TREE RRTPRN
*        **********************************************************
         LEVEL
*
         TREE DCMPLX-DCXADD-DCXDIV-DCXEXP-DCXLOG-DCXMPY-DCXSUB
         TREE GRAF1
         TREE HELP
         TREE ITITLE
         TREE LEXIT
         TREE MULE
         TREE STRPY
         TREE PCHEK-PZERO
         TREE PEQUAL
         TREE PPRN1-HOLLI
         TREE RCLAS-RTPRN0-XTRACT
         TREE RTEQU
         TREE RZERO
         TREE CPPRN
         TREE OSCALE
         TREE RPRINT
```

```
          TREE RTCMNT
*         DIRECTIVES FOR PLOTLIB
          TREE LEVEL2-NUMBER
          TREE GENGRD-NEWGRD-LINGRD-LOGGRD-FRAMES-LEVEL-IDFRAM-NAME
          END
```

## APPENDIX C. PROCEDURE INTLCAP2 FOR INTERACTIVE LCAP2

In Section 4.3 a block diagram for loading and executing Interactive LCAP2 is presented. The PROC (procedure) to implement this is given below:

```
.PROC.INTLCAP,FF=#FILE.
REMARK.WOULD YOU LIKE TO RETRIEVE DATA FROM A PREVIOUS SESSION?
REMARK.TYPE (Y OR N):
REPLY,SW1.
IFE,SW1.EQ.TRUE,DR.
REMARK.
REMARK.HAVE YOU ATTACHED THE PERMANENT FILE CONTAINING YOUR DATA?
REMARK.TYPE (Y OR N):
REPLY,SW2.
IFE,SW2.EQ.FALSE,DA.
REMARK.
REMARK.GO ATTACH THE PERMANENT FILE USING TAPE30 AS THE LOCAL
REMARK.FILE NAME, THEN START OVER BY TYPING LCAP2.
EXIT.
ENDIF,DA.
ENDIF,DR.
OFFDF.
ATTACH(ABS,8ABSINTLCAP2,ID=9487)
REQUEST(TAPE31,*PF)
REQUEST(PLOT,*PF)
ONDF.
ABS.
OFFDF.
DISCONT OUTPUT.
SET,R1G=0.
ATTACH(ZZ,8GENPROC,ID=9487)
COMMENT. ZZ WILL CREATE PROC1 AND PROC1 WILL SET R1G=1 IF BATCH
COMMENT. JOB TO CDC 176 FOR HARDCOPY IS REQUIRED.
ONDF.
ZZ.
OFFDF.
DISCONT OUTPUT.
ONDF.
BEGIN,PROC1,TAPE38.
OFFDF.
IFE,R1G.EQ.1,DR1G.
ONDF.
BATCH,TAPE39,INPUT.
ENDIF,DR1G.
LIBRARY.
RETURN,TAPE31,PLOT,ZZ,ABS.
ONDF.
REVERT.
```

This file is cataloged as permanent file 8INTLCAP2 with ID=9487.

APPENDIX D. PROGRAM GENPROC FOR INTERACTIVE LCAP2

In Section 4.3 program GENPROC which is used for post processing the outputs of Interactive LCAP2 is described. The code for this program is given below:

```
      PROGRAM GENPROC(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT,TAPE38
     +,TAPE39,TAPE89)
C     PROGRAM FOR POST PROCESSING OF INTERACTIVE LCAP2 JOBS.
      CALL CONNECT(5LINPUT)
      CALL CONNECT(6LOUTPUT)
      IUNIT=89
      REWIND IUNIT
      READ(IUNIT)NPLOTS,NSTORE,NDETRM
C
C     IF USER HAS REQUESTED HARDCOPY PLOTS, NPLOTS WILL BE NON-ZERO
C
C     IF USER HAS STORED DATA (EXECUTED STORE OPERATION), NSTORE
C     WILL BE NON-ZERO
C
C     CALL JBADGE(BADGE)
C     CALL JPGMR(PGMR)
C
C     CREATE PROCEDURE FOR
C     (1) CATALOGING DATA FILE CREATED BY STORE OPERATION
C
C     (2) CATALOGING PLOT FILE CREATED BY HARDCOPY REQUEST
C
C     CREATE PROCEDURE AND WRITE TO TAPE IUNIT1
      IUNIT1=38
      REWIND IUNIT1
      WRITE(IUNIT1,9000)
 9000 FORMAT(*.PROC,PROC1,P1=#FILE.*)
      IF(NSTORE.EQ.0)GO TO 20
C     ***************************************************
      WRITE(6,8900)
 8900 FORMAT(*0DATA FILE CREATED BY OPERATION STORE WILL BE CATALOGED*
     +,/* FOR YOU. ENTER 7 CHARACTER WORD TO BE USED AS FILE NAME:*)
      READ(5,8910)WORD
 8910 FORMAT(A7)
      WRITE(IUNIT1,8920)WORD,BADGE
 8920 FORMAT(*CATALOG,TAPE31,8*,A7,*,ID=*,A5,1H.)
      WRITE(IUNIT1,8930)
 8930 FORMAT(*REMARK.*/
     +*REMARK.DATA FILE HAS BEEN CATALOGED. TO USE THIS DATA FOR A*/
     +*REMARK.FUTURE JOB, ATTACH THIS DATA FILE PRIOR TO USING*/
     +*REMARK.INTERACTIVE LCAP2 AGAIN. THE ATTACH COMMAND WILL BE:*/
     +*REMARK.*/
     +*REMARK.*,38H************************************** )
```

```
      WRITE(IUNIT1,8940)WORD,BADGE
 8940 FORMAT(*REMARK.*,1H*,2X,*ATTACH,TAPE30,8*,A7,*,ID=*,
     +A5,4H.  * /
     +*REMARK.*,38H************************************** /
     +*REMARK.*)
C     **************************************************************
   20 CONTINUE
      IF(NPLOTS.EQ.0)GO TO 40
      WRITE(6,9022)
 9022 FORMAT(*0ENTER 7 CHARACTER WORD TO BE USED TO CATALOG PLOT FILES:*
     + )
      READ(5,8910)WORD
      WRITE(IUNIT1,WORD,BADGE
 9010 FORMAT(*CATALOG,PLOT,8*,A7,*,ID=*,A5,1H.)
      WRITE(IUNIT1,9015)WORD,BADGE
 9015 FORMAT(*REMARK.*/*REMARK.PLOT FILE CATALOGED ON CDC 835 AS*,
     +* 8*,A7,*,ID=,A5/*REMARK.(THIS FILE WILL BE PURGED UPON COMPLETIO
     +N OF HARDCOPY PLOTS)*/*REMARK.*)
      WRITE(IUNIT1,9018)
 9018 FORMAT(*SET,RIG=1.*)
   40 CONTINUE
      WRITE(IUNIT1,9020)
 9020 FORMAT(*REVERT.*)
      IF(NPLOTS.EQ.0)GO TO 500
C     CREATE CONTROL CARD FILE FOR BATCHING HARDCPY JOB TO CDC 176.
C     FILE SAVED ON TAPE IUNIT2.
      IUNIT2=39
      REWIND IUNIT2
      WRITE(6,9110)
 9110 FORMAT(*0ENTER LETTER TO DESIGNATE WHERE PLOT JOB WILL BE ROUTED T
     +O*/
     +* (D)BLDG D8 ,(F)BLDG A3, (J)BLDG 120, (K)BLDG A6, (N)BLDG D5*/
     +* TYPE (D,F,J,K OR N):*)
      GO TO 100
   90 PRINT*,"PLEASE TYPE (D,F,J, OR K):"
  100 READ(5,9120)RESP
 9120 FORMAT(A1)
      IF(RESP.EQ.1HD.OR.RESP.EQ.1HF)GO TO 130
      IF(RESP.EQ.1HJ.OR.RESP.EQ.1HK)GO TO 130
      GO TO 90
  130 DES=RESP
      WRITE(IUNIT2,9124)DES,PGMR
 9124 FORMAT(A1,A3,*,STMFZ,P3000,MS160000,ML40,T20.*)
      CALL JUNAME(NAME)
      CALL JO(JORDER)
      CALL JLOC(LOC)
      CALL JEXT(EXT)
      CALL JCCC(CCC)
      WRITE(IUNIT2,9130)NAME,BADGE,PGMR,JORDER,LOC,EXT,CCC
 9130 FORMAT(8HACCOUNT(,4X,A10,7X,A5,1X,1HI,A4,A6,1X,A8,5X,A5,1X,A4
```

```
     +,1H))
      WRITE(IUNIT2,9140)
 9140 FORMAT(*COMMENT. HARDCOPY JOB CREATED FROM INTERACTIVE LCAP2*)
      IF(DES.EQ.1HD)WRITE(IUNIT2,9145)PGMR
 9145 FORMAT(*MFLINK,OUTPUT,PRINT,PGMR=*,A3,
     +*,JOBCHR=D,CLASS=A,RJE=D8,MAIL=LCAP2PLT.*)
      WRITE(IUNIT2,9150)
 9150 FORMAT(*ATTACH(PLOTLIB,3FTNPLOTLIB)*/
     +*LIBRARY(PLOTLIB)*/
     +*FILE(PLTDATA,RT=S)*)
      WRITE(IUNIT2,9160)WORD,BADGE
 9160 FORMAT(*ATTACH(PLTDATA,8*,A7,*,ID=*,A5,*,ST=PF6)*/
     +*COPY,PLTDATA,PLOT.*)
      WRITE(6,9200)
 9200 FORMAT(*0DO YOU WANT HARDCOPY PLOTS PRODUCED IN D8 INSTEAD OF A3?
     + (Y OR N):* )
      READ(5,9120)RESP
      IF(RESP.NE.1HY)GO TO 140
      WRITE(IUNIT2,9165)
 9165 FORMAT(*HARDCPY,ST=IBMD8.*)
      GO TO 150
  140 WRITE(IUNIT2,9170)
 9170 FORMAT(*HARDCPY.*)
  150 WRITE(IUNIT2,9175)WORD,BADGE
 9175 FORMAT(*RETURN,PLTDATA.*/
     +*PURGE,XXX,8*,A7,*ID=*,A5,*,ST=PF6.*)
C     *******************************************************
  500 CONTINUE
      CALL DISCON(5LINPUT)
      CALL DISCON(6LOUTPUT)
      END
```

This program is compiled and cataloged as permanent file 8GENPROC with ID=9487.

☆ U.S. GOVERNMENT PRINTING OFFICE 1984-783-004/2126

END

FILMED

DTIC